

80  
24.



**UNIVERSIDAD NACIONAL AUTONOMA  
DE MEXICO**

**ESCUELA NACIONAL DE ESTUDIOS  
PROFESIONALES  
"ACATLAN"**

**"SISTEMAS MULTIUSUARIO Y REDES DE AREA LOCAL:  
UNA COMPARACION OBJETIVA"**

TESIS

QUE PARA OBTENER EL TITULO DE:

**LICENCIADO EN MATEMATICAS  
APLICADAS Y COMPUTACION**

Presenta

**EDGAR BENJAMIN VILLAGOMEZ VALDES**

Asesor de la Tesis

**FIS. MAT. JORGE LUIS SUAREZ MADARIAGA**

**TESIS CON  
FALLA DE ORIGEN**

Naucalpan, Edo. de Méx. Julio 1997.



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

**A Mónica, Oriana y mi bebé. Me impulsaron a cerrar el ciclo en esta etapa de mi vida profesional a través de amor, respaldo, paciencia y comprensión. Son la inspiración para alcanzar mis metas y brindo a ellos mis logros como ser humano. Este trabajo es por y para ustedes. ¡Gracias!**

**A Irene y Jorge. No tan sólo por darme la vida, sino por enseñarme los valores de la familia que son la herencia más importante y han hecho de mi un hombre de bien.**

**A mis hermanos. El estar tan lejos, y a la vez tan cerca, no es una tarea fácil. Gracias por su apoyo durante toda mi vida.**

**Al Fis. Mat. Jorge Luis Suárez Madariaga. Nunca perdiste la fé en que terminaría. Gracias por tu gula y soporte.**

## INTRODUCCION

<b>1. SURGIMIENTO DE LA TECNOLOGÍA NECESARIA PARA EL MUNDO ACTUAL</b>	<b>1</b>
1.1 El siglo XX. la revolución de la alta tecnología	1
1.1.1 Inicios	1
1.1.2 La microelectrónica	2
1.1.3 La era del microprocesador	3
1.2 Sistemas operativos	4
1.2.1 Evolución de los sistemas operativos	5
1.2.2 Ejemplos de sistemas operativos	7
1.2.3 Puntos clave en los sistemas operativos actuales	9
1.3 Tecnología para la productividad	11
1.3.1 Procesamiento distribuido	11
1.3.2 Migración de procesos	13
<b>2. SISTEMAS MULTIUSUARIO</b>	<b>15</b>
2.1 Introducción a los sistemas operativos multiusuarios	15
2.1.1 Conceptos de computadora virtual	16
2.1.2 Sistemas operativos multiusuario y multiproceso	17
2.1.3 Servicios del sistema	22
2.2 Arquitectura de UNIX	24
2.2.1 Panorama general de UNIX y sus principales componentes	24
2.2.2 Funcionamiento del <i>kernel</i> de UNIX	27
2.2.3 La arquitectura del <i>kernel</i> de UNIX	29
2.2.4 Sub-sistemas del <i>kernel</i> de UNIX	31
2.2.5 Organización de los procesos	31
2.2.6 Servicios de Entrada/Salida	33
2.2.7 Servicios de acceso a datos y archivos	35
2.3 Sistemas multiprocesador	37
2.3.1 Multiprocesamiento asimétrico	38
2.3.2 Multiprocesamiento simétrico	38
2.4 Procesamiento distribuido	40
2.4.1 Acoplamiento de sistemas distribuidos	40
2.4.2 Balanceo de cargas de trabajo a través de la migración de procesos	41
2.4.3 Procesamiento en paralelo	41
2.4.4 Sistemas de co-procesamiento	42
2.5 Modos de comunicación en UNIX	42
2.5.1 Comunicación asincrónica via modem	43
2.5.2 UUCP (UNIX to UNIX CoPy)	43

<b>3. REDES DE ÁREA LOCAL</b>	<b>47</b>
<b>3.1 Orígenes de las redes de área local (LAN)</b>	<b>47</b>
3.1.1 El mundo de las computadoras a principios de los 80's	47
3.1.2 El impacto de las computadoras personales	48
3.1.3 La comunicación entre computadoras personales	49
3.1.4 Los problemas al compartir archivos	49
3.1.5 Las redes de área local de los 80's	52
3.1.6 Hacia las redes de los 90's	53
<b>3.2 Arquitectura de las Redes de Área Local</b>	<b>55</b>
3.2.1 Procesamiento centralizado contra distribuido	55
3.2.2 Topologías de redes de área local	56
3.2.3 Principios del NetWare	61
<b>3.3 El modelo de las comunicaciones de red</b>	<b>64</b>
3.3.1 Los siete niveles del modelo OSI	66
3.3.2 El modelo OSI y las redes de área local	69
3.3.3 El modelo OSI y los dispositivos para la interconexión de redes	69
<b>3.4 Arquitectura de Novell NetWare</b>	<b>70</b>
3.4.1 Arquitectura del <i>kernel</i>	71
3.4.2 Protocolos de comunicación de NetWare.	71
3.4.3 TCP/IP, protocolo de comunicación para sistemas abiertos	73
<b>4. INTERCONECTIVIDAD</b>	<b>75</b>
<b>4.1 Conceptos básicos de comunicaciones</b>	<b>76</b>
<b>4.2 Métodos de transmisión y protocolos comunes</b>	<b>79</b>
4.2.1 Comunicaciones asincrónicas	79
4.2.2 Comunicaciones síncronas	80
<b>4.3 Redes de área amplia (WAN)</b>	<b>81</b>
4.3.1 Conmutación de paquetes	82
4.3.2 Conexiones virtuales	82
4.3.3 Componentes básicos	84
4.3.4 X.25	86
4.3.5 <i>Frame Relay</i>	87
<b>4.4 Ejemplo de implantación</b>	<b>89</b>
<b>5. CONSIDERACIONES EN LA TOMA DE DECISIONES</b>	<b>93</b>
<b>5.1 Ambientes gráficos</b>	<b>94</b>
5.1.1 Windows 3.11	94
5.1.2 Windows 95	96
5.1.3 X Windows y OSF/Motif	96
<b>5.2 Bases de datos relacionales</b>	<b>99</b>
5.2.1 Características de un DBMS	100
5.2.2 Soporte y herramientas para el sistema cliente	101

*Indice*

5.2.3 Ejemplos de DBMS	101
<b>5.3 Implantaciones híbridas</b>	<b>102</b>
5.3.1 SAP, ejemplo de implantación híbrida corporativa	104
<b>5.4 Puntos importantes de decisión</b>	<b>107</b>

**CONCLUSIONES**

**GLOSARIO DE TÉRMINOS**

**BIBLIOGRAFÍA**

## ***Introducción***

Muchas empresas, chicas, medianas y grandes, que requieren de la automatización de sus procesos de información para mejorar su productividad deben decidir qué tipo de herramientas utilizar para obtener el mejor resultado con la menor inversión. El propósito de este trabajo es el de dar una guía imparcial sobre la toma de decisiones en cuanto al tipo de sistema que se debe implantar de acuerdo a los requerimientos propios de la empresa, el tipo de información que se maneja y los recursos con los que se cuenta. Está dirigido a diseñadores de redes, consultores en informática, responsables de áreas de informática y público en general que deseen evaluar las ventajas y desventajas de la implantación de sistemas multiusuario y/o redes de área local.

Aunque ya es común el tema de la computación en nuestros tiempos, existen todavía personas y/o empresas que no saben que requieren de la asistencia de una herramienta electrónica para la solución de sus problemas. De acuerdo a la experiencia que he tenido a lo largo de nueve años de trabajo en el medio comercial proponiendo soluciones de informática e interconectividad a empresas de muy diversos ramos, no existen reglas para definir una solución global que se ajuste a todas las necesidades. Mientras las soluciones de redes de área local son buenas para muchas de las empresas, éstas tienen deficiencias que bien pueden ser cubiertas por las soluciones diseñadas a base de sistemas multiusuario y viceversa. También existen soluciones híbridas que, si bien son un poco más difíciles de implantar, abren la posibilidad de utilizar una gama más amplia de aplicaciones disponibles en el mercado.

Dado que ambos temas, sistemas multiusuarios y redes de área local, son muy amplios, este trabajo se limitará al análisis de las tecnologías líderes del

momento y no se tocarán temas básicos de computación, sin embargo, muchos de los conceptos poco usuales en el lenguaje común se podrán encontrar en el glosario de términos o deberá consultar el material que se lista en la bibliografía.

El trabajo está estructurado como sigue:

***Capítulo I. Surgimiento de la tecnología necesaria para el mundo actual.***

Presenta una breve historia del desarrollo de los sistemas de información a nivel *hardware* que sirvieron como base para los sistemas actuales.

***Capítulo II. Sistemas multiusuario.***

Historia y descripción técnica de los sistemas centralizados para acceso simultáneo a varios usuarios enfocado al sistema operativo UNIX, líder en este rubro.

***Capítulo III. Redes de área local.***

Historia y descripción de tecnologías de redes para que el lector comprenda los diferentes tipos de enlaces y el funcionamiento de la conectividad de computadoras personales.

***Capítulo IV. Interconectividad.***

Una vez que se tienen las bases de las tecnologías líderes de sistemas multiusuario y redes de área local, se explica la forma de conectar ambos para tomar sus principales ventajas y crear configuraciones óptimas para una implantación exitosa.

***Capítulo V. Consideraciones en la toma de decisiones.***

Se presentan los ambientes ideales para cada una de las tecnologías que han sido analizadas de forma que el lector pueda tomar en cuenta estos factores. También se presentan puntos clave que tienen como objetivo el brindar una guía de análisis rápida para evaluación.



## **1. SURGIMIENTO DE LA TECNOLOGÍA NECESARIA PARA EL MUNDO ACTUAL**

### ***1.1 El siglo XX, la revolución de la alta tecnología***

Desde la revolución industrial, el mundo no había experimentado una explosión de innovación social y tecnológica como lo está haciendo hoy. El desarrollo de nuevas tecnologías definitivamente ha impactado nuestra manera de trabajar y vivir a tal grado que nos hemos acostumbrado al cambio tecnológico continuo.

Las computadoras proliferan como nunca, son parte de nuestra vida cotidiana y existen millones de ellas alrededor del mundo. Es indudable que, en la actualidad, todos requerimos de información para poder realizar nuestras labores diarias. Entre mejor sea el flujo de la información que necesitamos, mejor es el aprovechamiento de nuestro tiempo y, a la vez, somos más productivos.

Es importante disponer de herramientas que incrementen nuestra productividad diaria. Para el volumen y complejidad actual de la información es indispensable la automatización del manejo de ésta mediante computadoras.

#### **1.1.1 Inicios**

La primera computadora electrónica digital construida en el mundo fue la Colossus, desarrollada en 1943 durante la segunda guerra mundial por la Gran Bretaña. Con el único propósito de descifrar los mensajes transmitidos por los alemanes, la Colossus fue una de las armas tecnológicas que permitió el triunfo de los aliados.

En 1946, la Universidad de Pennsylvania desarrolló la ENIAC, *Electronic Numerical Integrator and Calculator* (Integrador y Calculador Numérico Electrónico), una máquina que podía hacer mucho más que descifrar códigos; realizaba los miles de cálculos necesarios para poder analizar el desempeño de nuevas armas

y misiles. La ENIAC tenía grandes dimensiones, pesaba 30 toneladas y sus componentes incluían cerca de 70,000 resistencias y 18,000 bulbos con un costo aproximado de 2 millones de dólares.

La UNIVAC 1 fue la primera computadora comercial. Su primera instalación operacional se realizó en la General Electric y pronto la industria de cómputo comenzó su rápido ascenso.

### **1.1.2 La microelectrónica**

La microelectrónica ha sido el detonador en este constante desarrollo. Los costos de la capacidad de computación y memoria siguen bajando rápidamente gracias a sus adelantos.

Esta tecnología comenzó más o menos al mismo tiempo en que fue terminada la ENIAC. En realidad, fueron tres inventos clave los que impulsaron su desarrollo: el transistor, el proceso planar y el circuito integrado.

#### ***El transistor***

Fue desarrollado en las instalaciones de Bell Laboratories en los Estados Unidos. El proyecto de encontrar sustitutos para los componentes electromecánicos del sistema ENIAC, que causaban muchas fallas, dio como resultado el descubrimiento del efecto transistor, una débil amplificación de señales. Este invento tuvo tanto éxito que para los años cincuentas ya existían veinte empresas que producían diferentes tipos de transistores, todos ellos como componentes voluminosos.

#### ***El proceso planar***

En 1959, la compañía Fairchild Semiconductor produjo el primer transistor con procesamiento planar. Esta técnica consta de tres procesos:

1. Se oxida una pastilla de silicio (el principal ingrediente de la arena de playa), y luego se cubre con un material fotosensible, un fotorresistor.
2. Se fotografía un patrón sobre esta capa fotorresistente; patrón que se vuelve vulnerable a algunos productos químicos cuando se expone a la luz. El patrón fotografiado se graba a través de esa capa fotorresistente y el óxido que está debajo.
3. Se lava la capa fotorresistente para eliminar las impurezas.

Este proceso tiene la ventaja de que se pueden producir varias capas de transistores mediante el mismo procedimiento, disminuyendo el volumen y el tiempo en la producción de los transistores.

### ***El circuito integrado***

El procesamiento planar dio como consecuencia lógica la invención del circuito integrado, al colocar gran cantidad de transistores en un solo circuito impreso dentro de un pequeño espacio en base al silicio. Pronto se comenzó a implantar la producción a gran escala de estos circuitos integrados al incluir más funciones, bajando los precios por función integrada y aumentando su potencial. La principal limitante para los circuitos integrados era su rigidez ya que, al contrario de los sistemas como la ENIAC, no podían ser reprogramados una vez producidos.

#### **1.1.3 La era del microprocesador**

En 1969 fue inventado el microprocesador por los científicos de Intel. Este circuito integrado era una verdadera unidad central de proceso (CPU, *Central Processing Unit*) que no actuaba de manera rígida y preprogramada, sino que su lógica o respuesta podía ser alterada de acuerdo a los datos procesados. Al incluir dos *chips* (circuitos integrados) de memoria, uno para introducir y recuperar datos de el CPU y otro para proporcionar el programa para operar el CPU, el microprocesador adquirió un potencial comercial enorme ya que podría

entrar en aplicaciones de control una línea de producción, sistemas de semáforos y otras muchas aplicaciones antes nunca imaginadas.

Ahora encontramos computadoras en automóviles, lavadoras, relojes, sistemas de sonido, sistemas telefónicos y en muchos otros aparatos de uso diario. Todo esto gracias al gran desarrollo tecnológico que estamos viviendo, en donde la miniaturización de los componentes electrónicos ha rebasado la submicra (1 micra =  $10^{-6}$  metros) permitiendo que éstos diminutos dispositivos puedan colocarse en cualquier parte para controlar algún producto ó proceso de producción.

Los CPUs actuales encierran millones de transistores en un área inferior a dos centímetros cuadrados y son capaces de ejecutar millones de instrucciones por segundo. Para el próximo milenio podemos esperar cualquier cosa si el desarrollo tecnológico continúa con el mismo ritmo.

## **1.2 Sistemas operativos**

No obstante que los avances tecnológicos alcanzados hasta ahora en materia de microelectrónica han sido formidables, éstos no podrían ser aprovechados adecuadamente sin programas que puedan controlar sus recursos y que sirvan de enlace entre la computadora y la persona que la utiliza (*usuario*). Estas son las funciones básicas de un Sistema Operativo.

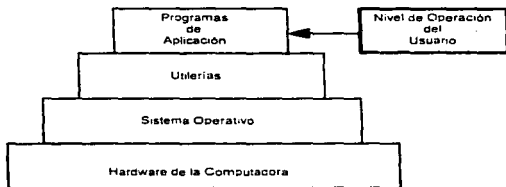
Adicionalmente a estas funciones, los sistemas operativos actuales cuentan con la mayoría de los servicios de: *utilerías para la creación de programas, ejecución de programas, acceso a los dispositivos de Entrada/Salida, control de acceso a archivos, control de acceso al sistema, detección de errores y monitoreo de actividad*, entre otros.

Para que un sistema operativo sea competitivo en el mercado actual, debe de cumplir los siguientes objetivos:

- *Conveniencia.* Que su uso sea práctico.
- *Eficiencia.* Que maneje los recursos del sistema de manera óptima.
- *Facilidad de actualización.* Que permita adición de nuevas funciones sin sacrificar la eficiencia en el servicio.

Para un usuario, una computadora se manifiesta en términos de aplicaciones. Este no tiene que saber cómo funciona la computadora por dentro para poder manipular información.

La Figura 1-1 muestra las capas en las que se divide una computadora desde el punto de vista de un usuario y qué papel juega el sistema operativo en la relación usuario/sistema.



*Figura 1-1 Capas de operación de una computadora.*

### **1.2.1 Evolución de los sistemas operativos**

Para entender el funcionamiento y alcances de los sistemas operativos actuales se debe dar un recorrido a través de sus diferentes etapas de evolución.

### ***Procesamiento serial***

Las primeras computadoras de los años 40's y 50's eran programadas de forma directa a través de *hardware* mediante el alambrado de tableros de control. No había un sistema operativo en *software* que controlara las funciones básicas del sistema.

En este tipo de computadoras existían dos grandes problemas: La administración de los tiempos para tareas y el tiempo requerido para poner a punto una aplicación.

Con el tiempo, estos sistemas fueron incorporando utilerías, como funciones comunes, manejadores de dispositivos de entrada/salida y *linkers* para hacer el procesamiento serial más eficiente.

### ***Procesamiento por lotes***

Este concepto se creó para tratar de minimizar el tiempo muerto en el que los sistemas, de alto costo, no procesaban nada. El primer sistema operativo de procesamiento por lotes (de hecho el primer sistema operativo) fue desarrollado a mediados de los 50's por General Motors en una IBM 701.

Estos sistemas implantaron un elemento de *software* llamado *monitor* que se encargaba de procesar segmentos de programas y datos del usuario en forma de bloque. De esta manera el usuario ya no tuvo acceso directo a los recursos de la máquina. La manera en que el usuario introducía datos y programas a la computadora era a través de tarjetas perforadas o cintas magnéticas.

Para que una computadora pudiera realizar procesamiento por lotes era necesario implantar algunos mecanismos de *hardware* como *la protección de memoria, un temporizador, instrucciones privilegiadas y manejo de interrupciones*.

Un mejoramiento al procesamiento por lotes fueron los sistemas operativos de procesamiento por lotes con capacidades de multitareas (la ejecución de varios procesos simultáneamente), en el cual varios usuarios podían obtener sus resultados al mismo tiempo, incrementando la productividad de la computadora pero todavía a través del uso de tarjetas perforadas o cintas magnéticas.

### **Sistemas de tiempo compartido**

Con el avance en la implantación de sistemas por lotes con capacidad de multitareas, el siguiente requerimiento lógico fue el de interacción directa de varios usuarios simultáneamente a través de terminales de video.

Esta implantación creó nuevos retos: la protección de datos entre usuarios para que no interfirieran unos con otros. La solución a este reto permitió varios de los mejores logros en la historia de los sistemas operativos modernos, como los siguientes:

- Mejoramiento en el manejo de procesos.
- Administración de memoria.
- Seguridad y protección de la información.
- Administración de recursos y calendarización.
- Estructura del sistema.

### **1.2.2 Ejemplos de sistemas operativos**

A continuación se presentan tres ejemplos de los sistemas operativos que han tenido gran relevancia y representatividad en la era de la computación.

#### **OS/2**

Para comprender la esencia del OS/2 es necesario hacer un poco de historia desde principios de los 80's. En 1980, Microsoft liberó el MS-DOS 1.0, un sistema operativo diseñado para trabajar en procesadores 8086 de Intel. Este

sistema operativo *monousuario* y *monotarea* ocupaba poca memoria. Con el rápido desarrollo de los procesadores, del 8086 pasando por el 80286 y hasta el 80386, el MS-DOS también evolucionó hasta la versión 4.0 en 1989 con el inconveniente de que solo utilizaba las ventajas de rapidez del procesador y no las características de direccionamiento de memoria y disco que estos procesadores brindaban.

Para estas fechas Microsoft e IBM lanzarían al mercado un sistema operativo que tomaría todas las ventajas de los procesadores actuales: OS/2. Este sistema operativo *multitarea* para ambientes monousuario incluía una interface gráfica que permitía al usuario la elección de opciones mediante menús y el manejo de ventanas para ejecutar diferentes aplicaciones simultáneamente.

La operación en modo multitarea permite la ejecución de más de una aplicación al mismo tiempo, ventaja que ha sido aprovechada en el procesamiento cooperativo junto con las minis y los mainframes, al repartir una aplicación para ejecutarse en más de un sitio a la vez.

### **UNIX System V**

El siguiente capítulo explicará ampliamente el desarrollo y las aplicaciones de este sistema operativo, sin embargo, vale la pena resaltar algunos puntos.

Este sistema operativo multiusuario/ multitarea es muy robusto debido a la portabilidad (ha sido modificado para gran cantidad de plataformas de *hardware*), seguridad y gran solidez que le han dado décadas de desarrollo. Su componente principal, y tal vez su principal aportación a la computación, es el *kernel* o núcleo. Este es el corazón del UNIX y provee las funciones básicas del sistema operativo como la ejecución de programas y el control de los dispositivos de la computadora. Alrededor del *kernel* se encuentran las aplicaciones del usuario y



la interface del usuario hacia el compilador C, que permiten la ejecución de aplicaciones sin que el usuario deba ser un experto.

### **NetWare**

Este sistema operativo de red ha tenido un crecimiento enorme en los últimos 10 años. Un capítulo posterior detalla su funcionalidad y confiabilidad. La clave de su éxito ha sido la facilidad de uso y la compatibilidad con aplicaciones con MS-DOS. A pesar de su corta edad, NetWare ha solidificado sus bases con firmeza, permitiendo que las aplicaciones corporativas sean confiables y con buen desempeño. Ha sido tan grande su aceptación por los usuarios y fabricantes de *hardware* que la industria de cómputo ha vuelto los ojos hacia esta tecnología y la ha convertido en la principal plataforma de aplicaciones gráficas y de escritorio.

#### **1.2.3 Puntos clave en los sistemas operativos actuales**

Los siguientes puntos son de gran importancia en la creación y desarrollo de la mayoría de los sistemas operativos modernos:

- **Concurrencia.** Existen dos temas principales para los sistemas operativos modernos: multiprogramación y procesamiento distribuido (ambos serán explicados ampliamente en capítulos posteriores). Para estos dos temas, y también para la tecnología de los sistemas operativos, es fundamental el término de concurrencia. Cuando múltiples procesos son ejecutados simultáneamente, o concurrentemente, normalmente aparecen conflictos entre ellos por la asignación de recursos de la computadora. En el desarrollo de los sistemas operativos se deben implantar mecanismos que aseguren la solución de conflictos para evitar caídas del sistema.
- **Manejo de memoria.** Un buen sistema operativo es aquel que aprovecha los recursos de la computadora de manera que su uso sea eficiente. la memoria principal de una computadora es uno de sus elementos esenciales. Normal-

mente, una optimización en el uso de la memoria (junto con un buen procesador) permite mejoramiento en el desempeño total de la computadora. Existen mecanismos de *hardware* y *software* que implantan algoritmos de carga de programas, realojamiento, paginación y segmentación, todos ellos mejoran el rendimiento del equipo.

- *Calendarización.* Una buena programación de los procesos, en términos del momento y el orden en que deben de ejecutarse, permite un mejoramiento en el desempeño total del equipo.
- *Manejo de dispositivos de Entrada/Salida.* El desarrollo en la tecnología produce dispositivos de entrada y salida cada vez más complejos como impresoras con calidad fotográfica, monitores de alta resolución, CD-ROMs y modems de alta velocidad. Para dar soporte a estos dispositivos, y a las aplicaciones que hacen uso de ellos, un sistema operativo debe ser capaz de controlarlos de manera eficiente.
- *Manejo de archivos.* Una buena estructura de datos es aquella que realiza una organización física y lógica de los archivos en los dispositivos de almacenamiento masivo en forma confiable y segura. El sistema de manejo de archivos en un sistema operativo es responsable de mantener la información accesible en todo momento y en el menor tiempo posible.
- *Redes y Procesamiento Distribuido.* El conectar computadoras a una red se ha vuelto una tarea común en estos días. Para poder hacer esta conexión, el sistema operativo debe de contar con utilerías que agilicen esta tarea. Una vez que una computadora está integrada a una red, puede ejecutar parte de una aplicación específica y contribuir a la distribución de procesos a través de la red.

- **Seguridad.** Cualquier sistema central que permita el acceso a más de un usuario remoto a la vez, es vulnerable a los intrusos. La seguridad es un tema muy importante, se debe tener en cuenta la integridad y confidencialidad de los datos contenidos en un sistema. Es necesario que un sistema operativo contemple mecanismos de seguridad que impidan a los intrusos el acceso al sistema.

### **1.3 Tecnología para la productividad**

Con la creciente disponibilidad de computadoras personales y minicomputadoras de bajo precio, se ha iniciado una tendencia hacia el *procesamiento distribuido de datos*. Esto, a través de la tecnología actual, permite a procesadores, datos y otros aspectos del procesamiento de datos estar dispersos en toda una organización e implica el particionamiento de todo el poder de cómputo hacia diferentes sitios geográficos.

#### **1.3.1 Procesamiento distribuido**

Existen muchas aplicaciones que requieren de datos de diferentes servidores de archivos. Esta información normalmente se encuentra dispersa geográficamente y es necesario implantar mecanismos de integración de información. Adicionalmente, estas aplicaciones se basan normalmente en grandes volúmenes de datos por lo que es necesario procesarlas localmente antes de intentar una integración. Este es el tipo de aplicaciones que implantan el procesamiento distribuido.

Varios factores tecnológicos son los que han permitido la implantación:

#### **Arquitectura de comunicaciones**

Normalmente se basa en *software* que soporta redes de computadoras. Provee compatibilidad con aplicaciones distribuidas como correo electrónico, transferencia de archivos y acceso remoto de estaciones de trabajo. Cada

computadora tiene sistema operativo independiente y es posible interconectar diferentes sistemas operativos siempre y cuando todas las computadoras utilicen la misma arquitectura de comunicaciones.

### ***Sistemas operativos de red***

Las instalaciones en donde existen computadoras de aplicaciones de red, estaciones de trabajo y servidores de archivos son ideales para la implantación de procesamiento distribuido. Los servidores proveen servicio o aplicaciones a toda la red, como almacenamiento de archivos y administración de impresoras. Cada computadora tiene un sistema operativo independiente. Los módulos de acceso a la red son sólo un complemento de los sistemas operativos locales que permite a las estaciones de trabajo interactuar con los servidores. Los usuarios saben de antemano que los recursos son compartidos y aceptan esta relación. Típicamente, es usada una arquitectura de comunicaciones común para soportar estas aplicaciones de red.

### ***Sistema operativo distribuido***

Es un sistema operativo compartido por las estaciones de trabajo de la red, que es visto por los usuarios de manera centralizada y les proporciona acceso transparente a recursos de diversos servidores. Estos pueden apoyarse en la arquitectura de comunicaciones para realizar funciones básicas de comunicación; las funciones de comunicación avanzada están incorporadas en él para garantizar la eficiencia en la funcionalidad.

La tecnología de la arquitectura de comunicaciones ha sido bien desarrollada y es soportada por todos los fabricantes de productos relacionados con procesamiento distribuido. La primera aplicación comercial fue *SNA (System Network Architecture)* desarrollada por IBM en 1974. Los sistemas operativos de red que contemplan el procesamiento distribuido han sido desarrollados más recientemente y aún incorporan mejoras a su funcionalidad.

### **1.3.2 Migración de procesos**

La migración de procesos es la transferencia de procesos de un sistema a otro cuando el primero se encuentra saturado. Esto ha sido motivado por la necesidad de balancear las cargas de trabajo entre los diferentes sistemas que conforman una red.

Las razones por las que es necesario implantarla son las siguientes:

*Compartir las cargas de trabajo.* Es necesario mover procesos de un sistema con demasiada carga de trabajo hacia un sistema subutilizado para mantener un buen rendimiento general de la red. Se debe ser muy cauteloso al implantar algoritmos de movilización de procesos ya que mientras más haya, mayor transmisión de datos habrá en la red y esto puede causar retraso en el tiempo de respuesta para el usuario.

*Rendimiento en las comunicaciones.* Los procesos que requieren de interacción constante del usuario deben mantenerse en el sistema más cercano a él para reducir costos de transmisión y tráfico en la red. Si un proceso realiza algún análisis de información de una base de datos de mayor tamaño que él, entonces debe ser movido al servidor en donde residen los datos.

*Disponibilidad.* Un proceso que requiere un largo período de ejecución en un sistema que va a dejar de funcionar (por mantenimiento, falla o apagado rutinario) debe moverse hacia un sistema que permita su terminación.

*Uso de características especiales.* Un proceso puede ser movido para tomar ventajas de recursos especiales de un sistema, ya sea *hardware* o *software*.

## 2. SISTEMAS MULTIUSUARIO

### 2.1 Introducción a los sistemas operativos multiusuarios

Los primeros años de la década de los 60's marcaron el nacimiento de los sistemas de *Time-Sharing* (tiempo compartido). Estos sistemas operativos fueron desarrollados en universidades como proyectos de investigación, utilizaban procesamiento por lotes y fueron la base para los sistemas actuales. El Instituto Tecnológico de Massachusetts (M.I.T.) desarrolló MULTICS como parte de un proyecto llamado MAC. Este sistema operativo marcó el inicio de la segunda generación y es ampliamente reconocido como el modelo de los sistemas operativos multiusuario y de tiempo compartido. El concepto de fragmentación de memoria en segmentos de longitud fija fue uno de los fundamentos para el desarrollo del MULTICS, que fue la primera implantación comercial del concepto de segmentos de memoria virtuales.

Los inicios de UNIX se remontan al año de 1969 con una minicomputadora PDP-7. Su principal aplicación fue la investigación para la programación y es por esto que su desarrollo fue simple y general. La naturaleza de esta implantación permitió la proliferación de herramientas de desarrollo y por esta razón su popularidad ha crecido desde el principio. Un análisis minucioso de su arquitectura revela la gran influencia de los conceptos básicos de MULTICS, como el método para compartir el tiempo de ejecución y sus esquemas de manejo de memoria. A través de los años, UNIX ha crecido en tamaño y en el repertorio de utilerías y herramientas. Ha sido portado a un gran número de computadoras y, en la actualidad, ha entrado en los ambientes de microcomputadoras, minicomputadoras y mainframes.

Dentro de las principales características de la arquitectura de UNIX se pueden listar las siguientes:

- Una arquitectura multiproceso.
- La habilidad de iniciar procesos en forma asincrónica.
- Un sistema de archivos jerárquico que permite el montaje de volúmenes.
- Compatibilidad de datos entre archivos, dispositivos y procesos.
- Selección de comandos a través de interacción con el usuario.

### **2.1.1 Conceptos de computadora virtual**

Los sistemas operativos multiusuario implantan este concepto y, desde el punto de vista del usuario, son una imagen de la computadora principal. Comparten recursos como discos duros, memoria, líneas de comunicación y, lo más importante, el CPU. El diseño de un sistema operativo de tiempo compartido permite brindar a cada usuario una o más imágenes de la computadora central al implantar algoritmos que permiten compartir el tiempo de CPU y los recursos adicionales.

El desarrollo de los conceptos de computadoras virtuales también provee la facilidad de portar aplicaciones de una implantación de UNIX a otra, ya que son casi insensibles al cambio de *hardware*. Este nivel de portabilidad es la clave de la popularidad de las aplicaciones desarrolladas para las diferentes implantaciones de UNIX.

### **Procesos**

UNIX provee recursos del sistema a los usuarios y programas a través de procesos. Un ambiente de máquina virtual consta de memoria, prioridades, una terminal para la interacción con el usuario y acceso compartido a otros dispositivos del sistema. Este conjunto de recursos están asociados a un proceso. Cada proceso es único y UNIX le asigna un número de identificación

(PID - *Process Identification*). Un proceso puede generar subprocesos para completar su tarea, puede terminar su propia existencia y puede ponerse en modo de hibernación para reiniciar su ejecución al cumplirse una condición predeterminada. Un ambiente multitarea se alcanza al coordinar varios procesos simultáneos en donde cada uno ejecuta una tarea diferente.

### **2.1.2 Sistemas operativos multiusuario y multiproceso**

Los sistemas operativos multiusuario manejan un cierto número de máquinas virtuales. Estas realizan las principales funciones en la administración de recursos, a saber:

- Uso del procesador (CPU) en una base de tiempo compartido.
- Administración de la memoria principal y secundaria.
- Administración del almacenamiento masivo.
- Control de dispositivos de entrada/salida.

Debido a la naturaleza de las máquinas virtuales, éstas brindan una serie de beneficios como los siguientes:

- Capacidad de interactuar con el *hardware* para ejecutar un gran número de tareas simultáneas.
- Los procesos ejecutados en ellas son aislados y protegidos unos de otros para evitar conflictos.
- Tienen un espacio de memoria propio.
- Brindan el acceso a todos los dispositivos del sistema.

Como las máquinas virtuales controlan los recursos del sistema, una aplicación que se ejecuta solo tiene que coordinar los accesos a los recursos a través de ellas sin complicaciones. Esta facilidad permite que una aplicación



desarrollada para una versión de UNIX pueda ser ejecutada sin modificaciones en otra versión e, incluso, en otro tipo de *hardware* diferente.

Para alcanzar los objetivos de este capítulo, que es el de dar un panorama general de la arquitectura interna de UNIX, se hará referencia a los procesos mismos, sabiendo que se ejecutan en máquinas virtuales.

Una de las principales funciones de sistema operativo es la de controlar los dispositivos de entrada/salida, como discos duros, terminales, impresoras, líneas de comunicación y otros periféricos. Una implantación eficiente es aquella que atiende a un dispositivo sólo cuando éste lo solicita, evitando la pérdida de tiempo. El mecanismo utilizado frecuentemente es el procesamiento de interrupciones (*interrupt processing*), en donde todos los controladores de dispositivos utilizados en un sistema tienen la capacidad de enviar una señal de interrupción indicando que el dispositivo requiere de atención. A cada interrupción de dispositivo se le asigna un nivel de prioridad para su atención. Si a un dispositivo se le asigna una prioridad alta, el sistema operativo suspende otros procesos, temporalmente, para poder responder ésta petición.

### **Conceptos de tareas, procesos y usuarios**

Una *tarea* es la ejecución secuencial de código de instrucciones. Una aplicación sencilla ejecuta código de instrucciones en forma secuencial. Por otro lado, las aplicaciones que ejecutan tareas en paralelo, por ejemplo, una aplicación que imprime mientras se comunica a un sistema remoto, ejecuta instrucciones separadas para cada tarea. Todas estas secuencias de instrucciones son parte del mismo programa y son controladas por el código de ejecución principal.

Una tarea se ejecuta dentro de un *proceso*. Un proceso puede ser, por lo tanto, monotarea o multitarea. Un sistema operativo que es diseñado para servir a un usuario o aplicación se le llama monousuario. Una implantación mas complicada es la de sistemas operativos que pueden asignar procesos a dos o mas usuarios y se les llama multiusuarios, de hecho, cada usuario puede generar subprocesos para ejecutar múltiples tareas al mismo tiempo.

### ***El uso del CPU y la administración de procesos***

El CPU es el principal recurso de un sistema. Ejecuta tanto programas del usuario como instrucciones del sistema operativo para manejar los recursos y programar procesos. El sistema operativo emplea un procedimiento ordenado para coordinar al CPU con el propio sistema operativo, los periféricos, aplicaciones de usuarios y llamados al sistema hechos por las aplicaciones del usuario. Existen algunos métodos utilizados por el sistema operativo para realizar esta labor. Los mas utilizados son:

- ***Secuencial.*** El sistema operativo realiza solo una tarea en cada momento.
- ***Barrido por evento.*** Es un gran avance con respecto al anterior ya que soporta múltiples tareas o usuarios. La programación de tareas es realizada por una serie de prioridades preestablecidas. Una tarea que ha sido marcada para ser el próximo evento a ejecutarse se programa y se ejecuta hasta que termina. Para tareas que requieren de mucho tiempo de ejecución, este método es poco eficiente ya que el tiempo de respuesta a un usuario puede ser muy largo. Aún con esta desventaja, este esquema ha sido muy utilizado por su facilidad de implantación y la rapidez para aplicaciones específicas.
- ***Manejo por interrupciones.*** Este método es una mejor implantación que el anterior ya que provee la capacidad de una rápida respuesta a las interrupciones de dispositivos.

- **Manejo de interrupciones por suspensión temporal.** Este método es utilizado en sistemas con aplicaciones de tiempo real críticas. Su principal característica es que, como su nombre lo indica, suspenden temporalmente una tarea de menor prioridad para dar paso a la ejecución de una de mayor prioridad. El sistema operativo UNIX entra en esta clasificación aunque solo suspende tareas derivadas de aplicaciones de usuarios y no de tareas del sistema mismo.

### **Administración de memoria**

La administración de uno de los principales recursos de un sistema, la memoria, es un concepto complicado en un sistema operativo multiusuario, ya que los requerimientos de los usuarios y del sistema operativo generalmente exceden a la memoria física disponible. El método mas utilizado para solucionar este problema es el de la *memoria virtual*, o en otras palabras, el uso de una parte del disco duro como una extensión de la memoria principal instalada en el sistema. Entre los esquemas mas utilizados para este fin encontramos los siguientes:

- **Barrido (Swapping).** Los procesos completos que no están en uso son enviados temporalmente al disco duro para regresarlos cuando son requeridos para ejecución.
- **Paginación por demanda (Demand Paging).** Solo parte de los procesos que no se requieren son enviadas al disco duro. Estas fracciones de información son llamadas *páginas*. Cuando el proceso requiere de esta página, envía una petición de página faltante para que sea recuperada y ejecutada. En este esquema, todos los procesos, al menos parcialmente, permanecen en memoria en cualquier momento.
- **Paginación por demanda y barrido.** Es una mejor implantación a la anterior ya que, además de manejar páginas de memoria, tiene la capacidad de

enviar procesos completos al disco duro para liberar memoria física y pueda ser utilizada por una página que sea recuperada.

### **Administración del almacenamiento masivo**

La principal regla para las estructuras de directorios en sistemas operativos multiusuario es un arreglo multinivel o árbol invertido. Estas estructuras deben implantar algoritmos para compartir archivos (*file sharing*) y sistemas de protección de archivos (*file protection*) que permitan el uso simultáneo de datos y aplicaciones por dos o mas usuarios a la vez y que eviten el acceso a la información por usuarios no autorizados. Otro de los mecanismos que deben ser implantados en los sistemas operativos multiusuarios es el de la operación asincrónica de procesos y periféricos. Esta operación implica la reprogramación de las peticiones de datos a los dispositivos de almacenamiento masivo ya que se deben ejecutar dos tareas simultáneas: una para verificar la disponibilidad de espacio en la memoria principal y otra para recuperar los datos del disco duro sin haber, necesariamente, dependencia entre ambas tareas.

### **Administración de dispositivos de Entrada/Salida**

Los dispositivos de Entrada/Salida (también llamados de E/S), incluyendo los dispositivos de almacenamiento masivo, son recursos compartidos que son utilizados por varios procesos de usuarios. A diferencia de los dispositivos de almacenamiento masivo, la mayoría de los dispositivos de E/S son utilizados de manera exclusiva por un proceso de usuario hasta que se ha completado su ejecución, por ejemplo una impresora o una línea de comunicación. La administración y el manejo de las interrupciones son las tareas fundamentales de la administración de los dispositivos de E/S.

### **2.1.3 Servicios del sistema**

Como se ha visto anteriormente, la primera y mas importante función de un sistema operativo es la de administrar los recursos del sistema, que incluyen al CPU, memoria, almacenamiento masivo y otros dispositivos periféricos. La segunda función es la administración de los servicios del sistema.

#### ***Servicios de E/S***

Los servicios de E/S consisten del inicio, control, transferencia de datos hacia y desde los dispositivos, así como del término de la comunicación con los periféricos. Los servicios de E/S incluyen los siguientes:

1. Direccionamiento de dispositivos.
2. Los *buffers* del sistema para transferencia de datos.
3. Manejo de las interrupciones de los dispositivos.
4. Manejo de errores de dispositivos.
5. Control de los dispositivos de impresión y de almacenamiento en cinta magnética.

Los servicios de E/S permiten compartir los dispositivos entre los usuarios y proveen un procedimiento uniforme para acceder los diferentes tipos de dispositivos. El sistema operativo reconoce las características del dispositivo y optimiza los mecanismos de acceso hacia él.

#### ***Servicios de acceso a datos y archivos***

Estos servicios proveen a los usuarios y aplicaciones acceso al sistema de archivos (*File System*) en disco de una manera simple y generalizada. El usuario o la aplicación sólo deben saber la dirección lógica y no la dirección física de los datos que requieran. El servicio de acceso a archivos realiza la traducción del nivel lógico al nivel físico del disco.

Los servicios que proveen los sistemas operativos multiusuarios incluyen los siguientes:

- Creación y borrado de archivos.
- Acceso a archivos por un nombre simbólico (una dirección lógica y no una dirección física)
- Organización de archivos de grupo por usuario o por tarea en la estructura de archivos.
- Control de acceso y compartir datos.
- Confiabilidad e integridad de datos.
- Expansión del sistema de archivos (*File System*).
- Adición y eliminación de discos del sistema de archivos.
- Control de respaldos.

El sistema de archivos del disco es generalmente tratado como un gran servicio de E/S ya que provee muchas más funciones que un servicio de E/S convencional.

### **Servicios de red y comunicaciones**

Estos servicios comparten las líneas de comunicación entre los diferentes usuarios de uno o mas sistemas. Proveen el control de las señales entre los dispositivos de comunicación y manejan la conversión de modos de comunicación entre los diferentes protocolos. Se verán estos servicios detalladamente mas adelante.

### **Servicios de mantenimiento de sistema**

Estos servicios incluyen los siguientes:

1. Instalación e inicio del sistema.
2. Control de accesos de usuarios (identificación de usuarios y claves de acceso).
3. Verificación y reparación del sistema de archivos.

4. Administración de controladores de dispositivos.
5. Respaldo y recuperación de datos.
6. Inicialización de parámetros de las líneas de comunicación.

Los servicios de mantenimiento proveen la automatización de las tareas para el administrador del sistema. Son herramientas para el mantenimiento diario del sistema de manera centralizada y controlada.

## **2.2 Arquitectura de UNIX**

En este apartado se verá a UNIX desde dos perspectivas diferentes: la arquitectura de los diferentes módulos y la funcionalidad de cada uno de ellos. Es importante resaltar que la funcionalidad mostrará las capacidades y la interrelación de un módulo con otro, en cambio la arquitectura sólo mostrará los componentes.

Como se menciona en la introducción, no se tocaran detalles de conceptos básicos de computación ni de arquitectura básica de computadoras; sin embargo, algunos de los términos y conceptos se podrán encontrar en el glosario para una mejor comprensión.

### **2.2.1 Panorama general de UNIX y sus principales componentes**

Una computadora típica consta de *hardware*, sistema operativo y programas de aplicación. Un sistema operativo típico incluye un intérprete de comandos o una interface gráfica equivalente para que el usuario pueda comunicarse con el sistema operativo. La mayoría de los sistemas operativos contienen un editor de textos, un compilador de lenguaje, un *linker* y otras utilerías. También suelen tener una interface de programación para que el usuario pueda desarrollar aplicaciones que manejen los recursos del sistema.

El sistema operativo UNIX tiene, además de lo anterior, una característica que lo hace único: la interface de comandos del usuario no esta integrada en el sistema operativo. Esta interface viene separada como programas independientes que se ejecutan cada vez que se tecllea un comando y pueden ser cambiados como cualquier programa de aplicación. Esto permite reducir el uso de la memoria para el sistema operativo y dejar disponible este recurso para programas de aplicación. La Figura 2-1 describe la estructura básica de una computadora corriendo el sistema operativo UNIX.

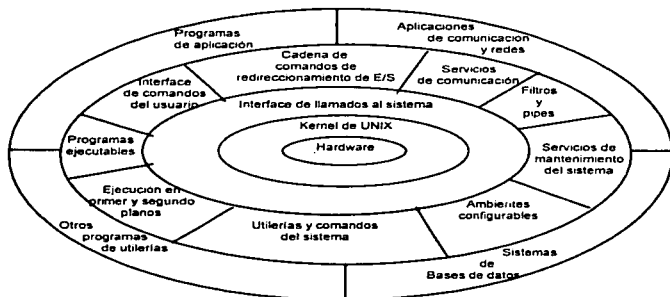


Figura 2-1 Arquitectura de UNIX

Los comandos o aplicaciones ejecutables de los usuarios interactúan con el *kernel* (núcleo) de UNIX a través de la interface de programación del sistema.

El sistema operativo UNIX contiene los siguientes cinco componentes:

1. El *kernel* (también llamado Sistema Operativo Básico).
2. La interface de comandos del usuario (generalmente llamada *shell*).



3. Los comandos y utilerías.
4. Los servicios del sistema.
5. La interface de programación.

Se verá una breve descripción de cada uno de los componentes anteriores para después profundizar en el *kernel* o núcleo de UNIX, ya que es el componente básico y el más complejo del sistema operativo.

#### ***La interface de comandos del usuario***

Este componente es un programa separado comúnmente llamado *shell* (o concha, ya que protege a las aplicaciones del usuario de las del sistema). Este componente tiene la capacidad de programación por si misma en un lenguaje parecido al C en su estructura. Existen varias versiones del shell dependiendo de las necesidades del usuario, normalmente hay más de una versión implantadas en cualquier variante de UNIX, como *csh*, *Bourne shell* y *kshell*. Algunos fabricantes de equipo, como SUN Microsystems, han implantado versiones gráficas al shell, tal es el caso del SUNtools, que manipula ventanas y elección de iconos y opciones en vez de la escritura de comandos, así como el uso de dispositivos apuntadores (*mouse*).

#### ***Comandos y utilerías de UNIX***

Estos comandos y utilerías, como el shell, son también programas separados. Son parte del sistema operativo pero no son considerados como parte del *kernel*. El número de comandos y utilerías depende de la variante de UNIX que se trate, pero se encuentra en un rango de 100 a 150.

#### ***Los servicios del sistema***

UNIX provee una serie de servicios básicos como la administración del sistema, reconfiguración del sistema y el mantenimiento del sistema de archivos. Adicionalmente, existen otros servicios como el de transferencia de

archivos (para copiar archivos entre los diferentes sistemas de una red local o remota), adaptación de parámetros del sistema y la reconstrucción del sistema operativo para incorporar manejadores del usuario.

El servicio más utilizado normalmente es el de administración del sistema. Este servicio realiza las siguientes actividades:

1. Creación y definición de cuentas de usuarios.
2. Definición de accesos a archivos y periféricos.
3. Administración del sistema de archivos.

### **La interface de programación**

Este componente provee a los usuarios acceso a las funciones de llamadas al sistema para manipular recursos como memoria, disco y periféricos.

#### **2.2.2 Funcionamiento del kernel de UNIX**

El *kernel* de UNIX, como se mencionó anteriormente, siempre ha sido reconocido como el componente básico del sistema operativo que soporta tanto programas del usuario como utilerías. Sus principales funciones son:

1. Programación de procesos.
2. Administración de memoria.
3. Administración de los dispositivos.
4. Administración de archivos y discos.
5. Manejo de la interface de llamadas al sistema.
6. Manejo de la interface de la consola del operador.

Todas estas funciones están distribuidas en una serie de módulos dentro del *kernel*. Las utilerías y los comandos de UNIX no son consideradas parte del *kernel*. Como se vió en la Figura 2-1, las capas que están más cercanas al

centro (*hardware*) están más protegidas del acceso del usuario. El *kernel* puede ser visto en un modelo funcional como lo muestra la Figura 2-2.

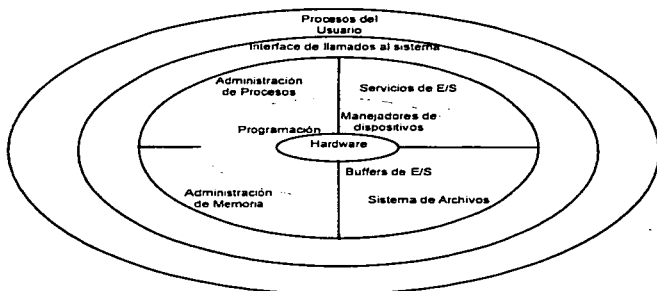


Figura 2-2 Modelo funcional del kernel de UNIX

Entre más cercana del centro esté la capa, mayor interacción tendrá con el *hardware* y, por lo mismo, mayor control tendrá sobre las actividades de programación de procesos y el manejo directo de la memoria para enviar y recibir datos. La siguiente lista muestra con mayor detalle las funciones del *kernel*:

1. Representación, programación y envío de procesos.
2. Asignación y desasignación de memoria.
3. Manejo de interrupciones.
4. Control a bajo nivel de dispositivos.
5. Administración de almacenamiento en disco.
6. Sincronización de procesos y comunicación entre procesos.

Cada vez que un proceso es cargado a memoria y empieza a ejecutarse, un segmento de memoria es reservado para el programa y sus datos. Adicionalmente, la memoria principal se requiere para *buffers*, bases de datos del sistema, entre otros. En la capa de los servicios del *kernel* se provee el mapeo entre las peticiones de las aplicaciones del usuario y las acciones de los manejadores de dispositivos. La capa de llamados al sistema convierte las aplicaciones de modo usuario a modo protegido del *kernel* para que el programa pueda invocar rutinas del *kernel* para realizar funciones del sistema. En la capa exterior se alojan los procesos que corren dentro del shell, comandos de UNIX, utilerías y programas de aplicación. Estos no tienen acceso directo al *kernel*, todos los accesos son canalizados a través de la interface de llamados al sistema.

### 2.2.3 La arquitectura del *kernel* de UNIX

Como se vió en la sección anterior, la arquitectura de UNIX se forma a través de capas de funcionalidad. Adicionalmente, contiene dos capas de protección contra errores de programación hechos por el usuario que pudieran afectar la ejecución de otras aplicaciones de usuario o algún componente del sistema operativo. Las capas cercanas al centro son llamadas *espacio del kernel* y las capas exteriores se les llama *espacio del usuario*.

#### ***Espacio del kernel***

El espacio del *kernel* es el espacio en memoria donde los servicios del *kernel* son suministrados a través de procesos. Cualquier proceso que ejecute código de *kernel* en este espacio es conocido como *proceso de kernel*. Un proceso de usuario que hace un llamado al sistema y comienza a ejecutar código de *kernel* se convierte en un proceso de *kernel*. Se dice que un proceso opera *en modo de kernel* cuando ejecuta código de *kernel*.

### **Espacio del usuario**

El espacio del usuario es el área de memoria en donde los procesos del usuario corren. Este espacio comienza justo arriba del *kernel* e incluye el resto de la memoria disponible. Un proceso que utiliza este espacio es conocido como *proceso de usuario* y se dice que está operando en *modo de usuario*.

### **Flujo de datos entre el espacio de usuario y el espacio de kernel**

Ya que el espacio de usuario y el espacio de *kernel* se encuentran alojados en áreas de memoria diferentes, el esquema de cómo mover datos entre ellos es importante. La mayoría de las implantaciones de UNIX contemplan el flujo de la siguiente información:

1. Información del proceso, como los archivos abiertos, los directorios accesados y los llamados al sistema.
2. El apuntador hacia la *tabla de procesos*, conteniendo la información requerida para programas de un proceso.
3. Tabla descriptora de archivos del usuario con información de los archivos que abre el proceso.

### **Puntos de entrada al kernel de UNIX**

Existen tres vías de acceso al *kernel* como lo muestra la Figura 2-3.

1. *Llamados al sistema* hechos por aplicaciones del usuario o por utilerías del sistema.
2. *Requisición de servicios de hardware* hechos por los manejadores de dispositivos.
3. *Condiciones de error*.

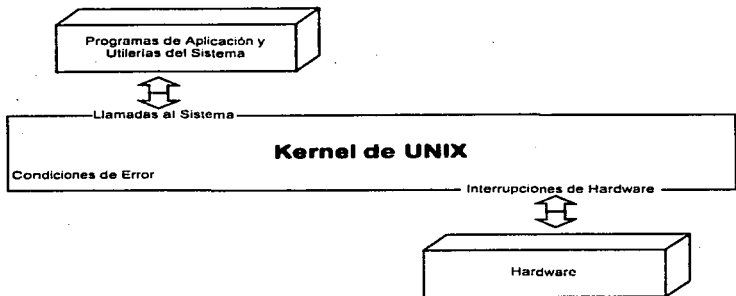


Figura 2-3 Puntos de entrada del kernel de UNIX

#### 2.2.4 Sub-sistemas del kernel de UNIX

El propio *kernel* de UNIX se encuentra dividido en módulos o sub-sistemas que proveen las funciones principales del sistema operativo. Cada uno de los sub-sistemas debe ser capaz de manejar sus propios recursos y poder comunicarse con los otros módulos. Estos sub-sistemas son los siguientes:

- Proceso 0 (el programador de procesos y CPU)
- Servicio de despacho de llamadas al sistema.
- Sub-sistema de servicio de archivos de nivel *kernel*.
- Módulo de servicios de E/S de nivel *kernel*.
- Sub-sistema de disco.

#### 2.2.5 Organización de los procesos

Los sistemas UNIX utilizan procesos para repartir los recursos del sistema entre las diferentes tareas. Cada aplicación o programa corre utilizando uno o

más procesos. A cada proceso que se genera se le asignan recursos dependiendo de su prioridad en el sistema y su habilidad de manejar estos recursos. Los sistemas UNIX utilizan *memoria virtual* para manejar gran cantidad de procesos. Los procesos que no son requeridos inmediatamente son puestos en el *área de swap* (barrido). El área de *swap* normalmente se genera en el momento de instalación de UNIX y reserva parte del disco para este fin exclusivamente. El *kernel* utiliza una combinación de la memoria física y el dispositivo de *swap* para manejar gran cantidad de procesos que no podrían ser mantenidos en la memoria física. Este método de optimización de recursos no podría funcionar sin la implantación de estrategias de programación de procesos eficientes. La siguiente tabla muestra una estructura típica de organización de procesos dentro de un sistema UNIX en funcionamiento:

UID	PID	PPID	C	STIME	TTY	TIME	COMMAND
Sched	0	0	3	Dec 31	?	0:01	swapper
Init	1	0	0	Dec 31	?	0:03	/etc/init
Root	301	1	0	10:16:45	co	0:21	-sh
Vtas	302	1	0	10:17:10	02	0:13	-chs
Proyect	313	1	0	10:17:13	05	0:05	/users/proyect/proyect
Conta	315	1	0	10:17:24	06	0:23	sh
Cron	329	1	0	10:17:34	?	0:04	/etc/cron
Almacen	340	1	0	10:17:45	09	0:12	/users/almacen/salidas
Lp	352	1	0	10:17:53	?	0:03	/usr/lib/lp/lpsched
Vtas	360	302	0	10:34:20	02	0:13	cat /etc/passwd
Conta	372	315	0	10:55:45	06	0:24	ps -ef

El proceso número 0 ejecuta el *swapper* y pertenece a *root* o administrador del sistema (PID=0). Este es el primer proceso creado por el sistema en el

momento del arranque. El segundo proceso creado, el proceso 1, ejecuta el programa */etc/init*. Todos los demás procesos son creados como sub-procesos a partir del proceso 1, el proceso *init* (inicial).

El proceso de *swap*, proceso 0, es responsable de programar los procesos. El programar los procesos incluye tres principales tareas:

1. Determinar qué proceso está listo para ser transferido hacia/desde el área de *swap* hacia/desde la memoria principal (*rutina sched*).
2. Transferir un programa que esta listo para su ejecución desde el área de *swap* (*rutina swapin*).
3. Programar tiempo de CPU al proceso transferido (*rutina swtch*).

Dado que todos los procesos tienen una prioridad determinada, un proceso de usuario puede ser interrumpido por otro proceso de usuario o de *kernel* con mayor prioridad. Un proceso de *kernel*, debido a que tienen siempre alta prioridad, no puede ser interrumpido por otro proceso de *kernel* ni de usuario.

El proceso *sched* (también llamado *swapper*) contiene el flujo operacional de los sistemas UNIX. La rutina *sched* utiliza dos banderas y un mecanismo de *inhibir/deshinibir* para determinar cuando un proceso está listo para ser transferido hacia/desde el dispositivo de *swap*.

### 2.2.6 Servicios de Entrada/Salida

Una característica muy particular de UNIX es la forma en la que trata a los dispositivos de E/S. Cada uno de éstos tiene un *archivo especial*, son parte del sistema operativo y tienen atributos de lectura y/o escritura como cualquier archivo de datos de usuario. La lectura o escritura en alguno de estos archivos provoca un acceso directo a su dispositivo asociado, incluso la memoria principal es accesible a través de un archivo especial. Los



dispositivos mas comunes, que pueden ser accedados por archivos especiales son discos, líneas de comunicación, impresoras, unidades de cinta, entre otros.

Los archivos especiales están organizados en el directorio */dev*. Se pueden hacer ligas de otros directorios hacia estos archivos para obtener facilidad de acceso a los dispositivos. La principal ventaja de este método es que no se requiere una interrupción de *hardware* para controlar el dispositivo y, por lo tanto, el manejo de éste es mas sencillo.

### ***Tipos de dispositivos de E/S***

Típicamente existen dos tipos de dispositivos:

- Dispositivos de Caracter (impresoras, terminales y otros que son orientados al manejo de caracteres).
- Dispositivos de Bloque (discos duros, unidades de cinta, entre otros.)

El sub-sistema de E/S provee la administración de estos dispositivos mediante el flujo de grandes cantidades de datos en un corto tiempo.

### ***Servicio de interrupciones***

Todos los dispositivos, incluyendo al reloj del sistema y el disco del sistema, se comunican con el *kernel* de UNIX vía interrupciones. La interrupción del reloj (que ocurre de 50 a 100 veces por segundo) ejecuta el programador de procesos para que los procesos tengan la oportunidad de competir por recursos del sistema muchas veces por segundo. Todas las interrupciones tienen una prioridad asociada a ellas. Dado que el sistema operativo utiliza frecuentemente al reloj y al disco, éstos tienen una alta prioridad y pueden interrumpir a otro proceso con menor prioridad.

## 2.2.7 Servicios de acceso a datos y archivos

Conocido como *FileSystem* en sistemas UNIX, el sub-sistema *servicio de acceso a datos/archivos* provee la facilidad de almacenamiento lógico y ordenado en dispositivos de almacenamiento masivo.

### Tipos de archivos

El sistema UNIX reconoce cuatro tipos de archivos: archivos de disco ordinarios (de texto o binario), directorios, archivos especiales y archivos de primeras-entradas-primeras-salidas (FIFO).

- Los ordinarios son los que utilizan la mayor capacidad del disco.
- Los especiales son usados para controlar los dispositivos de E/S.
- Los de FIFO son de longitud fija y se utilizan para pasar datos de un programa al otro.
- Los directorios se utilizan para organizar por grupos jerárquicos a los archivos.

### Estructura de directorios

La Figura 2-4 muestra la estructura típica de un sistema UNIX. Todo el árbol parte de la raíz o *root* denotada por "/". El sistema operativo utiliza como base a los directorios */dev*, */usr*, */etc*, */bin* y */user*.

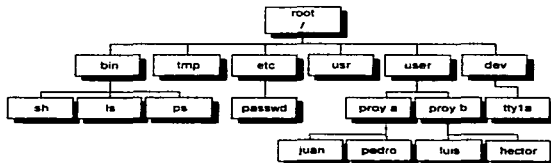


Figura 2-4 Estructura de archivos de UNIX

*/dev* -- Archivos especiales de dispositivos de E/S.  
*/usr* -- Comandos del sistema y archivos de programas.  
*/etc* -- Archivos de información del usuario.  
*/bin* -- Programas de utilidad.  
*/user* -- FileSystem montado con archivos del usuario.  
*/tmp* -- Archivos temporales para manejo de procesos.

### **Organización del disco e inodos**

Los archivos de los directorios asocian los nombres con números de *inodo*. El sistema UNIX construye y mantiene una lista secuencial de inodos en el disco. El número de inodo es su posición relativa en la estructura de inodos del disco. Cuando un archivo es accedido y abierto, el inodo asociado al archivo se copia en la tabla de inodos de memoria. El *kernel* hace referencia a este número para cualquier asunto relacionado con el archivo.

El disco, conocido como el FileSystem de UNIX, se divide en cuatro secciones o bloques. La primera sección, llamada *boot block*, es usada para el arranque de la máquina. La segunda sección, *superblock*, es usada para mantener el status de los usuarios en el sistema de archivos. La tercera sección es una serie de bloques de disco secuenciales que contienen registros de inodos secuenciales. La posición relativa del registro en la secuencia es el número de inodo. El sistema siempre ocupa el siguiente inodo disponible cuando se crea un archivo nuevo. La última sección consiste en los bloques de disco que contienen los datos reales de los archivos. Estos bloques pueden ser o no consecutivos ya que, debido a la creación y borrado dinámico de archivos, el espacio de disco es fragmentado y un nuevo archivo ocupa sólo los bloques disponibles.

### **2.3 Sistemas multiprocesador**

El sistema operativo UNIX fue creado para soportar múltiples usuarios y múltiples procesos en un modo de tiempo compartido. De esta manera, la carga de trabajo a nivel del *kernel* se incrementa considerablemente a medida que se incrementan los usuarios y los procesos, produciendo una baja sensible en el rendimiento total del sistema.

Algunas de las implantaciones de UNIX han considerado la migración hacia arquitecturas de *hardware* alternativas para incrementar el desempeño del sistema. El avance en las tecnologías de *hardware* han permitido el desarrollo de diferentes opciones de configuración para compartir la carga de trabajo que genera una aplicación al procesador y distribuir el acceso a una base de datos entre varios procesadores.

El *multiprocesamiento* está definido como la ejecución simultánea de varias tareas en múltiples procesadores dentro de un mismo gabinete, con el objeto de maximizar el rendimiento total del equipo. Bajo este esquema, el manejo de los diferentes procesadores (CPU) se realiza por demanda para balancear la carga de trabajo. Esta alternativa tecnológica no ha sido explotada del todo, aunque potencialmente puede brindar capacidad de proceso para más de 250 usuarios concurrentes. Los equipos multiprocesador impulsan la corriente informática del *downsizing* (la sustitución de minicomputadoras propietarias por equipos de gran poder basados en procesadores Intel), para obtener el mejor precio/rendimiento y la compatibilidad requerida para los estándares establecidos de los sistemas abiertos. En la actualidad existen dos tipos de multiprocesamiento que difieren, principalmente, en el tipo de aplicación para el que son utilizados.

### **2.3.1 Multiprocesamiento asimétrico**

En este tipo de multiprocesamiento existe, al menos, un procesador dedicado a una función especial (como el control de las interrupciones del sistema o de los procesos de E/S). Estos sistemas producen cuellos de botella, degradando el equipo cuando uno de los CPUs está a su máxima capacidad, mientras que los otros permanecen ociosos. Esta degradación es más notable cuando se utilizan aplicaciones que requieren de gran poder de cómputo, como las bases de datos. Cuando se tienen muchas terminales conectadas que utilizan aplicaciones "nobles", como procesadores de palabras, este tipo de sistemas tienen un rendimiento razonable.

### **2.3.2 Multiprocesamiento simétrico**

En contraste, los sistemas simétricos o, mejor dicho, los Sistemas Multiproceso Verdaderamente Simétricos (TSMP, por sus siglas en inglés) balancean la carga de trabajo entre todos los CPUs, ya que todos ellos tienen acceso directo a los recursos del sistema en cualquier momento, y de esta forma se obtiene un aumento casi lineal en el rendimiento a medida que se agregan CPUs, como lo muestra la Figura 2-5. Este rendimiento se alcanza al mantener la cola de espera de procesos en la memoria principal, compartida por los CPUs. Las aplicaciones son fraccionadas en pequeñas piezas y las peticiones de proceso son puestas en la cola de espera. Cuando un CPU se encuentra disponible, accesa a esta última y procesa la siguiente petición. Con este método ningún CPU permanece ocioso, al mantener un rendimiento estable aún con cargas de trabajo pesadas.

Fabricantes como Wyse (Series 7000i y 9000y), NCR (3450 y 3550) y Sequent (Symetry 200), adoptaron este tipo de tecnología y ofrecen el mejor rendimiento en este mercado, además de apearse a los estándares de Sistemas Abiertos.

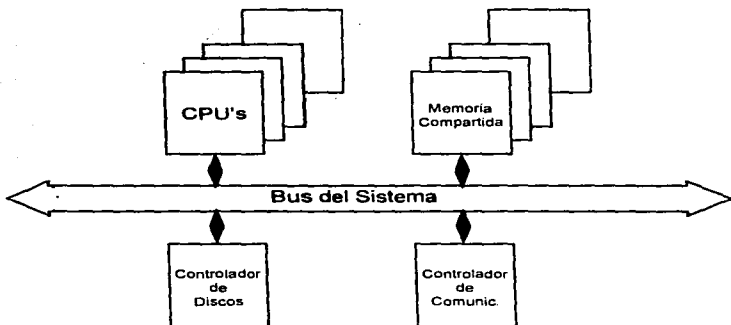


Figura 2-5 Esquema de la Arquitectura Multiprocesador

A principios de los 90's, Corollary organizó una "conferencia cumbre" sobre tecnología multiproceso. En esta reunión, altos directivos de empresas como InfoCorp, la misma Corollary, UNIX System Laboratories, Microsoft, Sun Microsystems (División *Software*), Novell, The Santa Cruz Operation e Intel, coincidieron en que la plataforma que tendrá mayor crecimiento en esta década será la de los equipos multiprocesador basados en tecnología Intel.

Si bien es cierto que la tecnología RISC ha tenido el apoyo directo de grandes fabricantes como Sun, I.B.M. y Hewlett Packard, las empresas desarrolladoras de *software* están dando un gran impulso a los sistemas operativos para sistemas multiprocesador basados en Intel. Tal es el caso de UNIX-MPX (The Santa Cruz Operation), BCS UNIX (AT&T) y Solaris (Sun Microsystems). Si además tomamos en cuenta que Intel tiene más de 15 años en el desarrollo de microprocesadores con gran éxito, es de esperarse que el Pentium tome el liderazgo tecnológico en esta década.

Es por ésto que los sistemas con este tipo de tecnología (Intel + TSMP) serán los que dominarán el mercado y tendrán el mejor apoyo de los fabricantes de *software*.

## **2.4 Procesamiento distribuido**

El término de *Sistemas Distribuidos* se emplea en cualquier aplicación en donde se realiza una repartición de tareas entre diferentes sistemas. Los sistemas actúan de manera tal que aparentan ser parte de un gran sistema debido a su coordinación de tareas y su capacidad de compartir recursos.

Esta sección introduce el concepto de acoplamiento de procesadores, migración de procesos de un CPU a otro en un ambiente de red con balanceo de cargas de trabajo y la adaptación de aplicaciones para funcionamiento en procesamiento en paralelo.

### **2.4.1 Acoplamiento de sistemas distribuidos**

Normalmente el término de *acoplamiento* se utiliza para identificar ambientes en los que los sistemas tienen una gran dependencia entre los diferentes CPUs, ya sea en una implantación Maestro/Esclavo o como punto-a-punto. En una implantación de tipo Maestro/Esclavo, en la cual un CPU (Maestro) se encarga de coordinar a otros CPUs (Esclavos) para optimizar los recursos disponibles, el rendimiento es excelente y, debido a la dependencia estrecha del Maestro con el Esclavo, el riesgo a una falla general del sistema es alto si un CPU falla. Por otro lado, en una implementación punto-a-punto la relación entre los diferentes CPUs no es tan estrecha, el rendimiento es relativamente bueno y el riesgo a una falla general disminuye considerablemente.

#### **2.4.2 Balanceo de cargas de trabajo a través de la migración de procesos**

El *migrar un proceso* implica realizar una copia de la imagen de un proceso, que se encuentra ejecutándose en un CPU, a otro en un ambiente de red de balanceo de trabajo. La migración de procesos de un CPU saturado a uno sub-utilizado ayuda a alcanzar balanceo de las cargas de trabajo y un rendimiento uniforme en la configuración total del sistema. La migración de los procesos es dinámica y no requiere ninguna modificación en las aplicaciones debido a que el rendimiento depende exclusivamente del *hardware* y del sistema operativo. Las extensiones del *kernel* de UNIX monitorean la actividad del CPU y los requerimientos de los procesos, para luego administrar los procesos que han sido migrados para asegurar que su ambiente de operación (archivos accedados, comunicación entre procesos, entre otros) continúen iguales.

#### **2.4.3 Procesamiento en paralelo**

Un sistema de procesamiento en paralelo, a diferencia de los de red con carga de trabajo compartida, trata a todos los CPUs como parte de un sistema, y las aplicaciones son desarrolladas para tomar ventaja de los diferentes CPUs distribuyendo tareas entre ellos. La mayoría de los sistemas de procesamiento en paralelo requieren de mejoras substanciales en el *kernel* de UNIX de manera que puedan repartir procesos entre los diferentes CPUs y proveer comunicaciones de alta velocidad para mantener la sincronía. La optimización de los recursos, en especial el acceso a la memoria principal compartida, requieren atención especial en este tipo de sistemas. Algunas implantaciones de UNIX incluyen un compilador de lenguaje C que determina los requerimientos de multitarea y multiprocesamiento de las aplicaciones y generan un archivo ejecutable que predetermina la distribución de procesos a través de los CPUs del sistema.



El procesamiento paralelo es una implantación especial del procesamiento distribuido. Este último se implanta desde un punto de vista corporativo, en cambio, el primero se implanta desde un punto de vista de sistema.

#### **2.4.4 Sistemas de co-procesamiento**

Este tipo de sistemas ha sido utilizado para aplicaciones de gráficas, comunicaciones, sistemas tolerantes a fallas, entre otros. En casi todas las implantaciones existe una gran dependencia de un procesador maestro y otro(s) esclavo(s). En el caso de un sistema tolerante a fallas, el procesador esclavo debe hacerse cargo de todas las tareas del maestro en caso de que éste falle. Normalmente, un procesador esclavo mantiene en forma independiente una copia del *kernel* del sistema operativo y se comunica con el maestro a través de memoria compartida. El procesador maestro es muy sensible a una falla del esclavo y es por eso que se deben implantar algoritmos de recuperación de fallas ya sea por *hardware* o a través de extensiones del *kernel*. Estas modificaciones deben ser diseñadas de manera tal que respondan rápidamente a alguna falla de un procesador y el sistema pueda seguir operando. Si la falla es no recuperable y/o el algoritmo no ha sido capaz de hacerlo, la ejecución del sistema operativo se detiene y el sistema falla.

#### **2.5 Modos de comunicación en UNIX**

El tema de comunicaciones se tratará con mayor profundidad en el capítulo de Interconectividad, sin embargo, para efectos de visualizar las ventajas de UNIX se verá brevemente.

Una de las principales facilidades que brindan los sistemas UNIX es la de comunicación con otros sistemas. Los servicios de *comunicación asíncrona* se encuentran incluidos en la base del sistema operativo y en sus extensiones.

El acceso a comandos a éste tipo de comunicación es inmediato y la definición de los parámetros es sencilla. Para que pueda funcionar con éxito se requiere de los puertos de comunicación adecuados. Un estándar para las comunicaciones asíncronas es el RS-232, que se encuentra en cualquier computadora personal de la actualidad.

Los puertos de comunicaciones RS-232 son los que normalmente se utilizan para conectar las terminales tontas que dan acceso a los usuarios a los recursos del sistema.

### **2.5.1 Comunicación asincrónica via modem**

Un *modem* es un dispositivo que permite la comunicación a través de líneas telefónicas a sitios remotos. Las ventajas que brinda este servicio de comunicación asíncronas, entre otras, son:

- Conectar sistemas que se localizan en sitios distantes.
- Permitir el acceso al sistema por usuarios remotos.
- Enviar y recibir información a dispositivos distantes (como impresoras)

Los modems se conectan al sistema por medio de líneas seriales asíncronas RS-232 y se configuran de manera que permitan la transferencia de información entre los dos puntos de enlace.

### **2.5.2 UUCP (UNIX to UNIX CoPy)**

El servicio UUCP permite a los sistemas UNIX comunicarse como parte de una red remota. Está conformado por programas que proveen las siguientes funciones:

- Transferencia remota de archivos (uucp).
- Ejecución remota de comandos (uux).
- Envío y recepción de correo electrónico entre puntos distantes (mail).

Se utiliza normalmente cuando la transferencia de datos es intensa y se requiere de comunicaciones de bajo costo. Normalmente se realiza a través de líneas telefónicas y el acceso puede ser por demanda o programado, permitiendo tanto llamadas de entrada como de salida. Un esquema típico se muestra en la Figura 2-6:

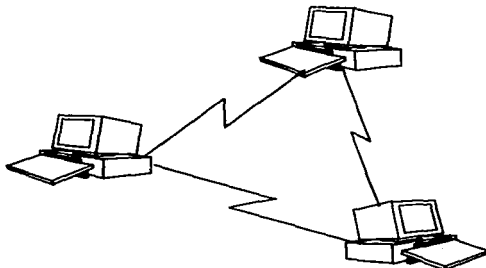


Figura 2-6 Conexión UUCP típica, a través de líneas seriales asincrónicas

El UUCP utiliza el método por lotes para controlar el tráfico de información, almacenando las tareas pendientes en una memoria temporal para procesarlas cuando el volumen de datos es suficientemente alto como para optimizar el tiempo de conexión y, por consecuencia, reducir el costo de los cargos por el uso de líneas telefónicas.

Cuando se ejecuta el comando *uucp* para realizar una transferencia de archivos, el servicio UUCP primero intenta localizar el primer dispositivo disponible para realizar esta tarea. Una vez que existe una línea disponible, el UUCP la bloquea para evitar conflictos con otros procesos e intenta establecer la comunicación con el sistema remoto. Si la conexión es exitosa, el UUCP del sistema que envía información ejecuta el comando *uucp* en el sistema remoto. Estos dos servicios negocian el protocolo de transferencia adecuado y se realiza el intercambio de datos. Si el sistema remoto tiene programado enviar información al sistema local, entonces éste se realiza, de otra manera se termina el enlace hasta que exista otra petición de transferencia. Si la conexión inicial no es exitosa, el UUCP intentará realizarla más tarde (normalmente en los próximos 60 minutos).

### **3. REDES DE ÁREA LOCAL**

Las redes de área local (LAN, por sus siglas en inglés), basadas en computadoras personales son muy diferentes, en arquitectura, diseño y aplicaciones, a las minicomputadoras o mainframes. Esto es debido a que su desarrollo tecnológico fue concebido en una era reciente de la evolución del procesamiento de datos y, adicionalmente, fueron creadas para cubrir requerimientos de usuarios que los sistemas multiusuarios no podían. Pero, aunque tienen diferentes aplicaciones y raíces, las redes de área local y los sistemas multiusuarios deben de enfrentar los mismos retos.

Ambos son *redes*, ya que conectan a muchos usuarios al mismo poder de cómputo, y deben de proporcionar la seguridad de que esta capacidad de procesamiento de datos sea útil, confiable y segura.

#### **3.1 Orígenes de las redes de área local (LAN)**

Las computadoras personales existieron cinco años antes que la primera red de área local basada en PCs apareciera en el mercado en 1980. Desde su creación, este tipo de redes tuvieron formas de operación diferentes a las de mainframes. Las siguientes secciones muestran los orígenes de las LANs.

##### **3.1.1 El mundo de las computadoras a principios de los 80's**

Cuando los circuitos integrados hicieron su aparición en los setentas, los fabricantes tradicionales de mainframes y minicomputadoras pensaron en crear sistemas con mejoras substanciales en desempeño y reducción de tamaño y precio. Pero lo que revolucionó la era de la computación fueron compañías como Apple y Commodore, que pensaron en desarrollar computadoras de escritorio buscando la masificación de esta tecnología. Es aquí en donde hacen su aparición las PCs, o Computadoras Personales.

### 3.1.2 El impacto de las computadoras personales

Las computadoras personales (PC) son cualitativamente diferentes a cualquier otro tipo de computadoras. Esta diferencia se basa en la sentencia "una persona, una computadora". Cuando la premisa de que una persona a la vez trabajará en la computadora se toma seriamente, entonces el diseño de estas computadoras se simplifica enormemente, cosa impensable en el mundo de los sistemas multiusuario de antes. De esta manera, los componentes obligatorios en sistemas operativos multiusuario como la seguridad, bloqueo de archivos y manejo de cuentas se desvanecieron por completo en los sistemas operativos para PCs.

El desarrollo de aplicaciones para computadoras personales provocó grandes ahorros de tiempo y esfuerzo. Este ahorro permitió la inversión de recursos para incrementar la facilidad de uso de estas aplicaciones mediante interfaces *amigables* y utilerías para el usuario. Para 1980, VisiCalc (la hoja de cálculo predecesora de Lotus 1-2-3) contaba ya con dos años de antigüedad y había mostrado a la industria de cómputo lo que realmente significaba *amigable*.

Los usuarios de las computadoras personales aceptaron inmediatamente esta simplicidad y comenzaron a demandar aplicaciones con plataforma estándar, cosa que en el mundo de las minicomputadoras y mainframes nunca se dió. Este deseo por la estandarización, o compatibilidad, dominó el mercado de la computación personal a mediados de los ochentas. Los fabricantes de aplicaciones apoyaron esta tendencia ya que impulsaban la simplicidad de desarrollo de aplicaciones y la penetración en el mercado.

Estos factores intervinieron para hacer a las PCs una industria llena de posibilidades de crecimiento, muy diferente a la de los mainframes y minicomputadoras. Los desarrolladores de Redes de Área Local para PCs

reconocen que esas fueron las bases fundamentales para la aparición de esta tecnología.

### **3.1.3 La comunicación entre computadoras personales**

La primera meta de las Redes de Area Local para computadoras personales fué la de compartir periféricos costosos (como discos duros e impresoras de alta velocidad). Los periféricos debían ser compartidos, pero esto rompía con la regla de "una persona, una computadora" a la que los usuarios de computadoras personales estaban acostumbrados. Adicionalmente, los discos duros fueron particionados para el acceso a varios usuarios y el sistema operativo de red fue modificado de tal manera que cualquier computadora dentro de la red pudiera "ver" al disco duro general como un disco duro local.

Las impresoras fueron compartidas mediante un manejador integrado al sistema operativo de red. Cuando los desarrolladores de sistemas operativos de red tomaron en cuenta las raíces de esta tecnología conservaron la compatibilidad de las aplicaciones creadas para el mundo *monousuario*. El ambiente operativo monousuario se mantuvo intacto y los usuarios pudieron migrar a esta tecnología sin problemas de capacitación y compatibilidad de aplicaciones.

### **3.1.4 Los problemas al compartir archivos**

Los servicios de disco en una red de área local son, en teoría, sencillos ya que lo "único" que realizan es permitir el acceso simultáneo de varios usuarios al disco duro central. Pero, tomando en cuenta que se deben mantener a los usuarios sin que exista conflicto de acceso a información privada entre ellos, los servicios de disco requieren de procedimientos complicados que realicen esta labor.

Existen muchos problemas que se deben resolver cuando se diseña un sistema operativo de red y se requiere especial atención en los servicios de disco, ya que éstos son la base para el buen funcionamiento general del sistema. Los problemas mas comunes son:

- *Controlar el CPU, la memoria RAM y los dispositivos de almacenamiento de manera que exista sincronización entre ellos.* Mediante el método que permite compartir tiempo de proceso (*time-sharing*) en un sistema operativo tradicional, un CPU reparte sus recursos entre muchas tareas. Se requiere de un programa que actúe como "supervisor" de procesos (como lo haría el *kernel* de UNIX) que vuelva a regresar todas las "piezas del rompecabezas" a su lugar de origen después de que un proceso ha terminado de ejecutarse. Este tipo de solución es regularmente implantado en minicomputadoras y mainframes. Para el sistema operativo de red NetWare 3.x, no se requiere este tipo de "supervisor" ya que la ejecución de los procesos se realizan en cada una de las estaciones de trabajo que lo necesita y que está conectada a la red, evitando la saturación del sistema central. Esta es la base operativa de NetWare: el sistema central únicamente hace las funciones de Servidor de Archivos, evitando el problema del manejo de *time-sharing*.
- *Compartir los archivos de datos entre los diferentes usuarios.* Se puede suponer, por ejemplo, que los usuarios A y B actualizan los datos del inventario en un sistema de red. El usuario A, que trabaja en el área de punto de venta, realiza un pedido de algún producto X. Para esto requiere consultar la existencia del producto (supongamos que existen 100 unidades) mientras el cliente decide cuantas comprará. Al mismo tiempo, el usuario B, que trabaja en el punto de entrada al almacén, consulta la existencia del mismo producto X (o sea, 100) y da entrada a 100 unidades más (para un total de 200) y actualiza la base de datos con esta información. Mientras tanto, el cliente del usuario A ya ha decidido comprar



10 unidades del producto X y también actualiza la base de datos con el total que su estación de trabajo ha calculado, es decir  $100 - 10 = 90$ . Pero, ¿qué pasó con la existencia real?, el usuario A sobre escribió la información que el usuario B había actualizado y el sistema tiene información incorrecta. A este tipo de situaciones se les llama el *estado de carrera*, en donde los usuarios "compiten" por actualizar la información del disco duro central y el ganador, paradójicamente, pierde. Para evitar este tipo de conflictos, se debe de permitir la escritura de información a un solo usuario a la vez, bloqueando la información a otros hasta que el primero que la accesa la actualiza. Este bloqueo puede ser a nivel archivo o registro: el de nivel de archivo es fácil de programar pero es ineficiente cuando la cantidad de usuarios se incrementa ya que el primero que accesa el archivo de datos tiene la exclusividad de uso y los demás usuarios deben de esperar su turno. El bloqueo a nivel registro es difícil de implantar pero tiene la gran ventaja de que no se bloquea todo el archivo, sino una parte de él, permitiendo el acceso a otras partes del archivo a los demás usuarios.

- El sistema de bloqueo soluciona el problema de el estado de carrera, pero crea uno nuevo: un *circulo vicioso*. Pongamos el mismo ejemplo de los usuarios A y B. Ya se ha implantado el sistema de bloqueo por archivo, por lo que nunca se repetirá el problema del estado de carrera. El nuevo sistema contempla dos archivos de datos, el de inventarios y el de clientes. Cuando el usuario A requiere hacer una venta consulta el archivo de inventario para verificar la existencia de un producto y, al mismo tiempo, consulta el de clientes para poder emitir una factura, de manera que mantiene ambos archivos bloqueados. El usuario B requiere hacer el mismo procedimiento. Ya que ambos usuarios realizan el mismo movimiento en la misma fracción de segundo, uno de ellos bloqueará el archivo de inventarios y el otro el de clientes. Como los dos usuarios requieren del archivo bloqueado por el otro, esperarán eternamente el

término de su transacción. Este problema se soluciona utilizando mecanismos avanzados de programación que actúen como árbitro.

Este tipo de complejidades hace difícil la implantación de un sistema operativo de tipo servidor de archivos, ya que los usuarios esperan que el rendimiento sea adecuado a su aplicación, que puedan compartir información, espacio de almacenamiento y periféricos. Los esfuerzos de los desarrolladores de sistemas operativos de red se enfocan directamente en mejorar estas capacidades para ofrecer al mercado productos altamente competitivos.

### **3.1.5 Las redes de área local de los 80's**

La gran penetración de las IBM PC como el estándar para las computadoras personales a mediados de los 80's marcó la pauta para los diseñadores de redes y fabricantes de PCs: El diseño de las redes de área local se haría en la plataforma de *hardware* con compatibilidad a PC y sistema operativo compatible con MS-DOS.

Los fabricantes se abocaron a la tarea de desarrollar productos compatibles entre sí, tales son los casos de Microsoft con MS-Net, Novell con NetWare, 3Com con el cable *Ethernet* delgado y Data General con ARCNet.

Al desarrollar estos productos, los fabricantes líderes vieron las grandes posibilidades de ofrecer algo más que compartir periféricos. Novell determinó que se requería un sistema operativo más robusto que ofreciera servicios de archivo, no sólo de servicios de disco.

Los *Servicios de Archivo* implicaban que el sistema operativo pudiera "decidir" en dónde almacenar los datos, y para esto se requería que el *hardware* del servidor fuera más "inteligente" y con mayor poder y, adicionalmente, que el sistema operativo de las estaciones de trabajo fuera modificado para una

mayor coordinación entre los diferentes componentes de la red. Este fue el primer gran paso hacia el concepto de *redes de alto rendimiento* de la actualidad.

El segundo paso en la evolución fué la independencia del *hardware*, permitiendo a los sistemas operativos de red funcionar con diferentes tipos de tarjetas de red, sin importar el fabricante, para que el usuario tuviera la libertad de opción. Esto no ha sido sencillo, ya que las computadoras no fueron diseñadas para ser dispositivos de comunicación y todavía existen problemas de incompatibilidad entre la relación computadora-tarjeta de red-sistema operativo.

Al alcanzar estos dos logros, las redes de área local se convirtieron en una forma práctica de interconectar un grupo de computadoras personales y establecer las bases para diseños más complejos.

### **3.1.6 Hacia las redes de los 90's**

Una red de área local que conjunta decenas de computadoras equivale, en términos de poder de cómputo, a una minicomputadora; una red que conjunta cientos de ellas equivale a un mainframe. Sin embargo, el poder de cómputo por sí mismo no lo es todo para la operación del sistema.

A mediados de los 80's, los diseñadores de redes enfrentaron dos retos: brindar la seguridad y confiabilidad equiparables a las minicomputadoras y mainframes, y crear métodos para interconectarlas con estos equipos grandes. El principal interés de lograr estos retos fue el de acceder la información que, hasta ese momento, era controlada en forma centralizada por las minis y mainframes

Por otro lado, los fabricantes de minis y mainframes diseñaban conectividad hacia el mundo de las PCs, pero con otro propósito: aprovechar la base instalada de redes, no para compartir la información, sino para utilizar las estaciones de trabajo como emuladores de terminales. Debido a esta diferencia en el desarrollo de tecnologías, la conectividad entre estos dos ambientes no tuvo un gran desarrollo en la década de los 80's.

Para lograr la confiabilidad requerida, los diseñadores de redes incorporaban protocolos de comunicación de red como TCP/IP, 3270, NFS (Network File System - creado por SUN para el ambiente UNIX) y otros, para incrementar su participación en el mercado mediante la creación de redes que fueran, cada vez mas, tolerantes a fallas.

A principios de los 90's, las redes de área local lograron la suficiente confiabilidad y versatilidad, el *downsizing* (la sustitución de equipos minis y mainframes propietarios por PCs de gran poder interconectadas) se convirtió en un término común. El número de estaciones de trabajo en una red se incrementó de unas cuantas decenas a cientos de ellas, y se convertían en la columna vertebral de la operación de las empresas, la especialización de los servidores se hizo cada vez más necesaria. Los servidores de impresión, de comunicaciones, de bases de datos, de procesamiento de imágenes, por mencionar algunos, fueron implantados para agilizar la operación diaria de las redes.

De esta manera, el concepto de redes (LAN) evolucionó a tal grado que pasó de ser un simple medio de compartir dispositivos costosos a ser el centro de operaciones de una empresa conectando una gran cantidad de computadoras personales con una gran variedad de mainframes.

## **3.2 Arquitectura de las Redes de Área Local**

Cada arquitectura de computadoras comienza como una solución a un problema específico. Esta solución es desarrollada con las herramientas disponibles en el momento del diseño para luego evolucionar hacia las arquitecturas que hoy conocemos.

### **3.2.1 Procesamiento centralizado contra distribuido**

El desarrollo de mainframes y minicomputadoras se inició cuando el poder de cómputo y las líneas de comunicación eran sumamente caras. Por ésta razón las redes fueron desarrolladas de naturaleza necesariamente jerárquica; el alto costo de las líneas de comunicación significaba que cualquier esfuerzo dedicado a la descentralización del poder de cómputo se enfocara siempre a la minimización de este rubro.

Las antecesoras de las redes actuales eran sistemas que corrían procesos por lotes y eran programadas a través de tarjetas perforadas o cintas magnéticas. Mientras el poder de cómputo crecía, los diseñadores de sistemas buscaban diferentes medios para procesar tareas y datos. El primer paso hacia la descentralización se logró con la implantación de un sistema operativo de tiempo compartido junto con terminales no controladas directamente por el CPU. Mas tarde, se agregó mayor inteligencia a los dispositivos involucrados en la red jerárquica (Controladores de Comunicaciones y de Segmento) para liberar al CPU y las líneas de comunicación de la carga de trabajo que ponía en riesgo la confiabilidad y seguridad de la red

Esta tarea de agregar inteligencia a los dispositivos de comunicación logró la disminución del tráfico con el consecuente incremento en la solidez de la implantación.

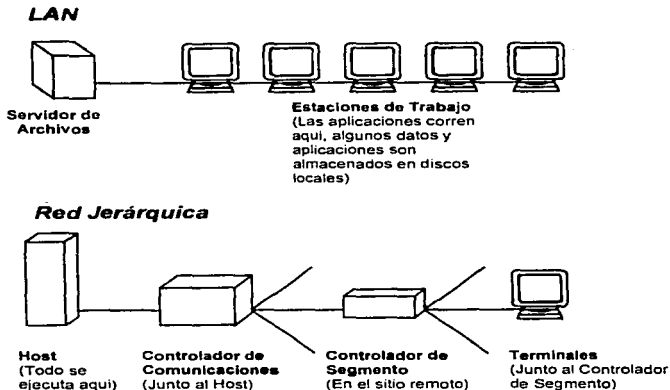


Figura 3-1 LAN y redes jerárquicas

### 3.2.2 Topologías de redes de área local

Para lograr la conectividad de las PCs y los servidores en una red de área local se requiere de varios dispositivos que puedan comunicarse entre si de manera coordinada y transparente. Una topología define la conexión física de los dispositivos en una red, de manera que hagan eficiente la transmisión de información. A continuación se describen, en orden histórico, las principales topologías de red.

**Estrella**

Es una topología de LAN en donde los nodos son conectados por líneas de comunicación a un concentrador central llamado *HUB* y de éstos hacia el servidor, como lo muestra la Figura 3-2. El depender de un punto de conexión central constituye un alto riesgo de falla. La adición de nodos es transparente para los usuarios ya que puede hacerse mientras el sistema está funcionando. La comunicación es eficiente entre los nodos y el servidor, no así la de un nodo con otro, ya que la señal debe pasar siempre por el servidor para retransmitirse al nodo de destino.

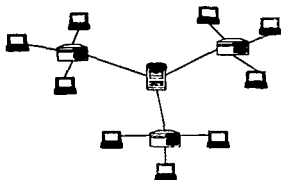


Figura 3-2 Topología de Estrella

**Bus**

En ésta topología de red, los nodos se conectan directamente a un canal de comunicación central. Un estándar de este tipo de topología es *Ethernet*. Los nodos tienen una dirección única a nivel mundial y no hacen la función de repetidor de señal, por lo que no retrasan el desempeño de la comunicación. La expansión no requiere adición de cable y, por su naturaleza, la falla de ningún nodo puede hacer fallar a la red. El esquema general se muestra en la Figura 3-3.

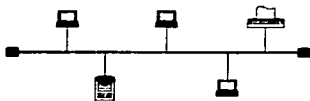


Figura 3-3 Topología de Bus

### Token Ring

Los nodos están conectados punto a punto formando un círculo, o anillo. Es una contribución de IBM a los estándares de topologías y está incrementando rápidamente su base instalada. Los nodos actúan como repetidores de señal y retransmiten los mensajes al contiguo. El control es distribuido, pero ahora, con muchos puntos de falla, cualquiera de los nodos puede interrumpir el funcionamiento de la red. La expansión también interrumpe la comunicación ya que el anillo debe romperse para insertar un nuevo nodo ya que existe alta dependencia en la continuidad de la señal como lo muestra la Figura 3-4:

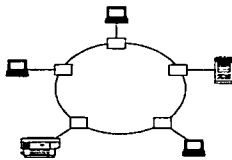


Figura 3-4 Topología Token Ring



Cada una de estas topologías utiliza un método de transmisión de datos, que a continuación se describen.

### **Polling**

Es el tipo de transmisión que utiliza la topología de estrella. El nodo central controla las comunicaciones de la red ofreciendo el mismo tiempo de acceso, uno a la vez. Es un método lento ya que el nodo central se mantiene siempre accediendo a cada nodo de la red en busca de información para transmitir, aún si éste no tiene nada o está apagado.

### **Contention**

Método utilizado por la topología de bus. El canal central de comunicación es "apartado" por una fracción de segundo cada vez que un nodo desea transmitir información. Una vez completada la transmisión, el canal se libera para que otro nodo pueda transmitir. Este método implementa mecanismos para detectar y solucionar *colisiones* de transmisión simultánea para asegurar el funcionamiento eficiente de la red.

### **Token Passing**

Método utilizado por la topología de Token Ring. Un *token* (segmento de bits) pasa a través de la red siempre. Si un nodo desea transmitir, éste inserta su mensaje al principio del segmento de bits y reinserta el token a la red. Debido a su velocidad de transmisión y su fácil manejo, este método es el que tiene una gran aceptación en las redes actuales.

Como último punto en las conexiones físicas de las redes de área local es el del medio de transmisión. La elección del tipo de medio depende de factores como volumen de transmisión, costo, facilidad de instalación, entre otros. A continuación se detallan:

### ***Par Trenzado (Twisted Pair)***

Existen dos tipos de cables de par trenzado: con y sin blindaje. El par trenzado sin blindaje (UTP, *Unshield Twisted Pair*) es conocido comunmente como cable telefónico. Se utiliza normalmente en instalaciones en donde ya existe una infraestructura telefónica y la implantación es sencilla, eficiente y económica. De hecho es el estándar en implantaciones de redes de área local. Por otro lado, el par trenzado con blindaje (STP, *Shield Twisted Pair*) es más caro y se utiliza en instalaciones que requieren protección adicional a fuentes de energía electromagnética que pudieran generar interferencia en la transmisión de datos.

### ***Cable Coaxial***

Este tipo de cable de cobre es el que regularmente se utiliza en la televisión por cable. Es más difícil de instalar que un cable de par trenzado ya que es más rígido. Sin embargo, es ideal para la transmisión de datos, ofrece un gran ancho de banda para grandes volúmenes de datos, alta velocidad de transmisión, inmunidad a las interferencias eléctricas y bajo nivel de error. Sus principales desventajas son la poca flexibilidad y que es propenso a fallas cuando no se instala correctamente.

### ***Fibra Optica***

Se constituye de finos filamentos de vidrio que conducen impulsos de luz a grandes distancias. Se requiere de dispositivos que conviertan los impulsos eléctricos a impulsos de luz para poder instalar este medio. Las principales ventajas de la utilización de la fibra óptica son: gran ancho de banda, inmunidad a interferencia eléctrica y flexibilidad de instalación. Su precio es superior a los demás medios pero ha ido bajando con el tiempo. Es ideal en la instalación del backbone (espiná dorsal) de redes de alta velocidad.

### **3.2.3 Principios del NetWare**

La premisa de la red de área local de Novell, NetWare, fue la idea del *procesamiento distribuido*. Mediante la adición de nuevas estaciones de trabajo a la red, se podría aumentar el poder de cómputo en toda la red en una relación directamente proporcional. De esta manera, los usuarios no notarían la degradación del sistema como tradicionalmente se presentaba en los sistemas del tipo de las minicomputadoras o mainframes.

Pero, no solamente se eliminó la degradación del poder de cómputo, los diseñadores de redes obtuvieron algo que jamás planearon: el mundo de las aplicaciones de las computadoras personales. El acceso a estas aplicaciones incrementaron el interés de los usuarios de redes por este ambiente, algo que ningún sistema operativo para PC en red había logrado. La clave: la compatibilidad de aplicaciones con el sistema operativo monousuario de mayor uso en el mundo, MS-DOS.

Los primeros diseños del NetWare se hicieron compatibles con los equipos del tipo CP/M y DOS. En ambos casos, la red aparentaba la presencia de discos duros locales en las estaciones de trabajo, que en realidad se alojaban en el servidor de archivos.

#### ***Estaciones de Trabajo***

Por el lado de las estaciones de trabajo o nodos (principales componentes de una red), el NetWare actúa como un supervisor de llamados al sistema para el acceso de disco duro. Si un llamado es para un disco local, NetWare permite su paso, sin cambios, hacia el controlador de discos locales. Si el llamado es para un disco de la red, este es interceptado y desviado hacia la tarjeta de red, como lo muestra la Figura 3-5.

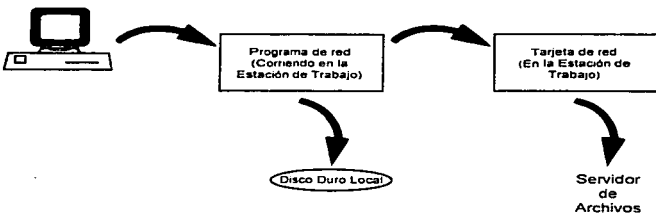


Figura 3-5 Llamados de la Estación de Trabajo

### **Servidores de archivos**

Una de las principales diferencias entre las redes de área local y las minicomputadoras/mainframes es el rol de la computadora central. En las instalaciones de minicomputadoras/mainframes, ésta ejecuta los programas de aplicación. En cambio, el sistema central de las redes sirve como una biblioteca de archivos, o mejor dicho servidor de archivos, para las estaciones de trabajo. Las aplicaciones son ejecutadas en cada una de las estaciones de trabajo y el servidor de archivos se encarga de pasar la información entre estos dos puntos en la medida que la aplicación lo requiere.

Las primeras implantaciones de los servidores de archivos para red fueron sistemas con arquitectura propietaria. Después de la introducción de la PC AT, muchos de los servidores de archivos fueron computadoras personales con procesador 286 ó 386. Con el desarrollo tecnológico y la creciente necesidad de procesar información, los servidores de la actualidad pueden llegar a tener miles de veces mayor poder de cómputo que los primeros equipos.

**Otros servidores**

El servicio de archivos es sólo una de las actividades involucradas en una implantación *cliente-servidor* típica. Una vez que la estación de trabajo obtiene información del servidor de archivos, la primera se convierte en *cliente* del segundo, que será el *servidor*. Esta relación da inicio al desarrollo de muchos otros servicios: servidores de bases de datos, servidores de comunicaciones, entre otros. Un solo servidor puede proveer de recursos a una gran cantidad de usuarios o estaciones de trabajo.

**Interconexión de redes**

Uno de los principales motivos del éxito de la industria de las computadoras personales fue el de la participación de empresas especializadas en la fabricación de los componentes opcionales para estos equipos. Para 1985, las empresas fabricantes de computadoras personales dejaron de producir sistemas operativos y, en cambio, empresas especializadas los elaboraban. Para las redes de área local las cosas se hicieron de manera semejante. El resultado fue una gran variedad de tarjetas, sistemas de cableado y topologías que fueron ofrecidas al público para 1987. La siguiente tabla muestra estos productos:

<b>Compañía</b>	<b>Tipo de Cable</b>	<b>Protocolo</b>	<b>Topología</b>
SMS	RG 64 Coaxial	Token Bus	Estrella Lineal
3Com	RG 58 Coaxial	CSMA/CD	Bus
Synoptics	Par Trenzado sin Blindaje	CSMA/CD	Bus
IBM	Par Trenzado con Blindaje	Token Ring	Anillo

La interconexión a través de puentes (*bridge*) hicieron fácil la comunicación entre redes de diferentes topologías. También permitieron el crecimiento modular de las redes, especialmente cuando el usuario requería comenzar con una red, por ejemplo, ARCNet de 5 usuarios y después implantar una red *Ethernet* de 15 usuarios para interconectarlas mediante un puente y de esta manera compartir los recursos de la inversión inicial.

Este tipo de crecimiento, relativamente fácil de implantar, permitió las redes incrementar su base instalada de una forma explosiva.

### ***Aplicaciones***

Muchas de las aplicaciones para redes son simplemente extensiones de ambientes monousuario instaladas en servidores de archivos y no en un disco duro local. Sin embargo, las que son desarrolladas específicamente para redes van más allá que la localización física de la aplicación. Entre más elaborada sea la aplicación, mayor ventaja tomará de las facilidades que un ambiente de procesamiento distribuido puede brindar como la capacidad del acceso simultáneo de múltiples usuarios a la misma aplicación. Estas capacidades de multiusuario, combinadas con el concepto de *cliente-servidor*, son la base de cualquier aplicación orientada a redes de área local. Los sistemas operativos de red, como el NetWare, proveen una gran cantidad de comandos para el bloqueo de archivos y/o registros que previenen el *estado de carrera* y el *circulo vicioso* que ya se han mencionado anteriormente.

### ***3.3 El modelo de las comunicaciones de red***

Para permitir a los sistemas comunicarse a través de diferentes líneas de comunicación se requiere de complejos procedimientos. Dentro de estos procedimientos, probablemente el concepto más importante sea el *protocolo*.

Como en la realeza, el protocolo dicta las reglas que se deben seguir para obtener un "buen" comportamiento

Si dos personas de habla hispana mantienen una conversación, seguramente podrán intercambiar ideas y opiniones de manera fluida y sin errores. Por otro lado, si dos diplomáticos, un Francés y un Chino, desean llegar a una negociación deberán utilizar traductores que conozcan el protocolo de ambos países para asegurar una traducción fiel y no haya malos entendidos.

El mundo de comunicación de datos es similar, se deben comprender diferentes protocolos de comunicación para poder asegurar un intercambio de información fluido y sin errores.

Para que existiera una estandarización de conceptos de protocolos de comunicación, los diseñadores de estas tecnologías como IBM, DEC y la Organización Internacional de Estandarizaciones (ISO por sus siglas en inglés) acordaron representar un modelo de las tareas de comunicaciones a través de niveles (*layers*) de manera comprensiva y estructurada.

Para IBM, esta estructuración dio vida a lo que hoy conocemos como SNA (*Systems Network Architecture*), DECNet para DEC y el modelo OSI (*Open Systems Interconnection*) para ISO. Debido a que las dos primeras organizaciones decidieron crear estos modelos para uso exclusivo, el único modelo que permitió la estandarización de las redes que hoy conocemos fue el OSI.

La gran fuerza del modelo OSI es la de no definir estructuras para un solo fabricante, si no como un modelo descriptivo a seguir para la estandarización en comunicaciones. De esta manera, los fabricantes de dispositivos de comunicación aseguraron la compatibilidad de sus productos con los de otros

fabricantes. Sorprendentemente, este mismo modelo fue aplicado con éxito en el mundo de las redes de área local, en donde los estándares fueron implantados a nivel de las estaciones de trabajo y no en el nodo central como había sido diseñado originalmente. La Figura 3-6 presenta el modelo OSI y los niveles que lo componen:

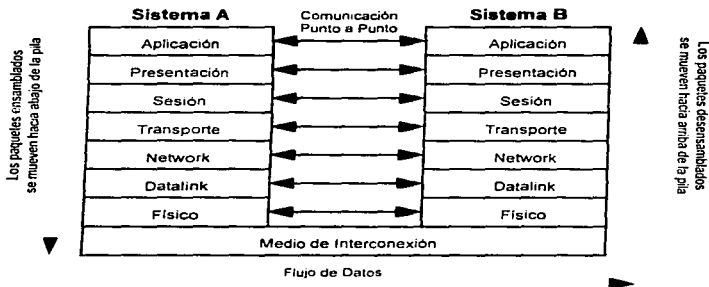


Figura 3-6 Modelo OSI

### 3.3.1 Los siete niveles del modelo OSI

El modelo OSI realiza la transmisión de datos de un punto a otro mediante la división del trabajo en siete pasos, que son arreglados secuencialmente en niveles (*layers*), y cada nivel contribuye al ensamblado de paquetes (*packets*) de información de manera que los datos contenidos en ellos siempre estén identificados con su procedencia y destino.



Los cuatro elementos de información que contienen los paquetes para su control son los siguientes:

- *Caracteres de inicio.* Informan al dispositivo receptor que el paquete está en camino.
- *Encabezado.* Contiene la información necesaria para identificar la procedencia y destino del paquete así como el tipo de que se trata (de datos o de control de red).
- *Datos.* Es la información del usuario.
- *Bits de verificación de errores y caracteres de fin de paquete.* Bits que verifican la integridad de la información una vez que ha sido recibida.

Todos los paquetes de información que son enviados pasan por un proceso de ensamble en los diferentes niveles por el sistema emisor. De manera complementaria, el sistema receptor desensambla el paquete en el orden inverso en el que fue ensamblado para su correcta interpretación.

Los niveles del modelo OSI son como sigue:

- *Nivel 1 o Físico (Physical Layer).* Este nivel genera los pulsos físicos necesarios (eléctricos u ópticos) para transmitir los datos desde la tarjeta de red hacia el sistema de comunicación. La interfase RS-232 es un ejemplo de un estándar a nivel físico. Este nivel no incluye el sistema de comunicación, pero sí incluye la conexión hacia él.
- *Nivel 2 o Datalink (Data Link Layer).* Este es el primer nivel que maneja paquetes como tales. Realiza el ensamble final (como los bits de corrección de errores) a los paquetes de salida y la primera revisión (como la integridad de datos) a los de entrada. Si se detecta un paquete con errores, entonces se descarta y se envía un mensaje de error al punto emisor.

- **Nivel 3 o de Red (Network Layer).** Este nivel realiza el ruteo de paquetes entre múltiples dispositivos. Si un paquete que es enviado debe viajar a través de ruteadores, *gateways* o *bridges* entonces es necesaria la intervención del nivel de Red, ya que se encarga de mantener actualizadas las tablas de ruteo y determinar la vía más corta para que el paquete llegue a su destino. Esto es esencial cuando el tráfico de la red es intenso y se requiere optimizar el uso del canal de transmisión. Algunos ejemplos de protocolos de comunicación que operan a este nivel son el IP (*Internet Protocol*) de TCP/IP y el IPX (*Internetwork Packet Exchange*).
- **Nivel 4 o de Transporte (Transport Layer).** El TCP (*Transmission Control Protocol*) de TCP/IP opera a este nivel. Este es un nivel de transición, ya que es último nivel que maneja ruteo y corrección de errores. Sirve muchas veces como respaldo al nivel de Red ya que puede controlar errores que éste último no contempla.
- **Nivel 5 o de Sesión (Session Layer).** El nivel de Sesión es el que mantiene las transmisiones en conexiones continuas. Realiza los procesos de establecer y terminar una conexión. En este nivel se presume que los paquetes contienen información correcta, por lo que no es necesario realizar labores de corrección de errores. Los protocolos *NetBIOS* y *SPX* operan a este nivel.
- **Nivel 6 o de Presentación (Presentation Layer).** La función de este nivel es confundida frecuentemente. No tiene relación con la presentación de información en la pantalla del usuario (de estas tareas se encarga el nivel de Aplicación). Su principal función es la de preparar los datos para que puedan ser utilizados por el nivel de Aplicación como, por ejemplo, la conversión de datos entre los formatos ASCII y EBCDIC.

- **Nivel 7 o de Aplicación (Application Layer).** Este nivel maneja la seguridad de sistema, la disponibilidad de recursos, transferencia de archivos y despliegue de información en las terminales.

### **3.3.2 El modelo OSI y las redes de área local**

La arquitectura de las redes de área local no se adapta fácilmente al Modelo OSI de una manera natural. El concepto de correr una aplicación en la estación de trabajo en lugar de hacerlo en un sistema centralizado implica que la estación de trabajo debe integrarse a la red a un nivel más alto del Modelo OSI. Todo lo que requiere la estación de trabajo es una ruta de acceso al servidor de archivos, es decir, se debe manejar el nivel 3 (network). Ninguno de los dos, la estación de trabajo y el servidor de archivos, requiere establecer una sesión (nivel 5) ya que lo único que se necesita es enviar y recibir paquetes. El nivel 6 sólo presenta a la estación de trabajo un sector virtual de disco y el nivel 7 se encarga de proveer los parámetros de seguridad cuando el usuario inicia la conexión a la red.

Esta arquitectura puede parecer demasiado sencilla para el Modelo OSI, y en realidad lo es, pero los diseñadores de las redes de computadoras personales visualizaron la necesidad de intercomunicar estas redes con otros tipos diferentes y decidieron preparar sus diseños con la compatibilidad requerida por un estándar ampliamente aceptado.

### **3.3.3 El modelo OSI y los dispositivos para la interconexión de redes**

Algunos dispositivos de *hardware* han sido diseñados especialmente para manipular paquetes en un nivel específico del modelo OSI. A continuación se presentan tres de ellos y el nivel OSI en el que operan:

- **Repetidores.** Son dispositivos de Nivel 1 (Físico), sólo amplifican la señal de cualquier paquete que pase a través de ellos.
- **Puentes.** Mueven paquetes a través de dos redes que operan bajo la misma topología de red, sin importar el protocolo de comunicación. Se analiza cada paquete y se determina si la dirección destino es local o remota. Si esta dirección es local, el paquete no pasa a través del *punto*, si es remota el paquete es enviado a la otra red. Los *puentes* operan en el Nivel 2 (*Data Link*), por lo que son "invisibles" para los dispositivos que operan en un nivel superior del modelo OSI. Su principal aplicación es la de reducir el tráfico de una red grande al partirla en dos pequeñas e interconectarlas mediante un *punto*.
- **Ruteadores.** Estos dispositivos operan en el Nivel 3 (Network) del Modelo OSI. Fueron diseñados para interconectar dos o más redes con diferentes topologías, siempre y cuando ambas utilicen el mismo protocolo de comunicaciones. Por ejemplo, un *punto* puede interconectar una red *Ethernet* con una ARCnet si las dos utilizan IPX como protocolo de comunicación. Por el otro lado, un *ruteador* puede interconectar redes aún si se están utilizando topologías de Estrella, Bus o Anillo en las diferentes redes. Un *ruteador* tiene la capacidad de elegir el mejor camino de un punto a otro de una serie de redes interconectadas para disminuir el tiempo de respuesta si es que existe más de una ruta disponible.

### 3.4 Arquitectura de Novell NetWare

El sistema operativo de red Novell NetWare tiene un diseño avanzado que permite el óptimo aprovechamiento de las arquitecturas de 32 bits en las PC's actuales. Integra un soporte de comunicación a los sistemas operativos líderes del mercado como MS-DOS, MS-Windows, Macintosh, OS/2 y UNIX, así como comunicación a través de Redes de Area Amplia (WAN), mediante módulos de *hardware* y *software* adicionales.

### 3.4.1 Arquitectura del *kernel*

Al igual que UNIX, Netware contiene un grupo de rutinas internas que controlan el sistema operativo. Los componentes del *kernel* de Netware son los siguientes:

- Ejecutor del sistema
- Calendarizador
- Protocolos nativos, como SPX/IPX y NCP
- Mecanismo de ruteo
- Administrador de memoria
- Sistema de archivos
- Administración de semáforos
- Conexiones virtuales a diferentes protocolos como TCP/IP.

El *kernel* puede ejecutar estos procesos mediante el calendarizador de tareas, que es un grupo de procesos que controla la ejecución de el resto de los procesos. Este determina qué tarea deberá de ejecutarse en cada momento, pero no cuánto tiempo debe de ejecutarse.

### 3.4.2 Protocolos de comunicación de NetWare.

#### **IPX**

El protocolo natural de comunicación de NetWare es IPX (*Internetwork Packet Exchange*), que no requiere de conexión ya que él maneja la transmisión de paquetes de una manera óptima. Opera a nivel 3 (network) y proporciona a las estaciones de trabajo y al servidor de archivos servicios de direccionamiento y ruteo entre diferentes redes.

El IPX se compone de las siguientes secciones: 32 bits de direccionamiento de red, un número de *host* de 48 bits y 16 bits para el número de *socket*. Estas tres secciones son indispensables para formar una dirección de internet completa e identifica a cada computadora dentro de una red.

El *software* que implanta IPX debe ser cargado en cada estación de trabajo que se desea conectar a la red IPX. En el caso del servidor de archivos, el IPX se encuentra interconstruido en el Sistema Operativo, permitiendo su comunicación con las estaciones de trabajo y otros servidores.

Otro de los principales servicios que brinda IPX es el de ruteo de paquetes entre redes. Un ruteador IPX completo se encuentra interconstruido en el servidor de archivos de NetWare y le permite ser el punto de enlace de dos o más redes. Un servidor de archivos debe tener una conexión (tarjeta de red) por cada red a la que vaya a transmitir paquetes. Para que un paquete llegue a su destino es necesario que el ruteador sepa el camino de acceso más cercano al punto de destino; para esto se requiere que cada uno mantenga una tabla actualizada de las direcciones de los demás ruteadores que interconectan a las diferentes redes. El mantenimiento de estas tablas lo realiza el RIP (*Routing Information Protocol*).

Para que una estación de trabajo corriendo DOS pueda tener acceso a una red IPX debe ejecutar el programa IPX.COM, que se genera mediante el módulo IPX.OBJ de Novell y el manejador de la tarjeta de red (que debe ser proporcionado por el fabricante de la tarjeta) con los parámetros correctos de configuración de *hardware*.

### **SPX**

El SPX (*Sequenced Packet Exchange*) es un protocolo orientado a conexiones que provee conectividad confiable en aplicaciones de red punto a punto. Trabaja a nivel de transporte y permite un intercambio de datos en forma secuencial eliminando errores en el flujo de datos y la duplicidad en la entrega de paquetes. Para la entrega de paquetes el SPX utiliza los servicios de IPX, que opera a un nivel superior en el modelo OSI.

El SPX encapsula sus datos y su información de control en la sección de datos del paquete IPX. La información de control contiene los siguientes elementos:

- **Control de conexión:** Controla los datos a través de la conexión.
- **Tipo de datos:** Identifica el tipo de datos que los protocolos de más alto nivel deberán de interpretar.
- **Identificación del sistema emisor:** Nombre único del sistema en la red.
- **Identificación del sistema receptor:** Igual que el anterior.
- **Número de secuencia:** Eliminan la duplicidad de paquetes recibidos.
- **Número de reconocimiento:** Igual que el anterior.
- **Número de asignación:** Informa al sistema emisor cuánto espacio queda disponible para el envío de paquetes.

### **NCP**

Es un protocolo utilizado para solicitar recursos y servicios. El NCP controla la interacción entre las estaciones de trabajo y el servidor de archivos mediante el establecimiento de conexiones entre ellos y el procesamiento de las solicitudes de servicios del servidor de archivos. Estas peticiones pueden ser lectura/escritura de archivos, impresiones y comunicaciones.

### **3.4.3 TCP/IP, protocolo de comunicación para sistemas abiertos**

IPX es un protocolo de comunicación en red muy eficiente para Netware, sin embargo, no es la mejor opción para toda clase de sistemas.

El TCP/IP (*Transmission Control Protocol/Internet Protocol*) es uno de los protocolos de red más popular en la actualidad. Se encuentra implantado en gran variedad de sistemas operativos, principalmente en UNIX. Fue desarrollado a principios de los 70s para el Departamento de Defensa de los Estados Unidos con la idea de permitir la conectividad de diferentes tipos de sistemas y varios tipos de medios de comunicación.

Existen tres razones principales por las que TCP/IP es muy popular:

1. *Es un poderoso protocolo para enlace de redes.* Permite a un sistema la conexión simultánea a múltiples medios de comunicación, extendiendo las posibilidades de conectar redes entre sí. TCP/IP puede configurarse para trabajar con casi cualquier tipo de medio de comunicación, incluyendo *Ethernet*, *Token Ring*, *X.25* y conexiones punto a punto.
2. *Ofrece gran conectividad.* Como resultado de la creación de la red del Departamento de Defensa de Estados Unidos (ARPANET), surgió Internet, una red mundial basada en TCP/IP. Muchas instalaciones adoptaron este protocolo ya que les daba acceso a Internet.
3. *Provee madurez y disponibilidad inmediata.* Lleva veinte años en el mercado y muchos usuarios ya se han acostumbrado a él. Existe en casi todas las variantes de sistemas operativos UNIX y muchos fabricantes de *software* desarrollan productos de conectividad que lo incluyen como protocolo de comunicación de red estándar.

La relación de TCP/IP con el modelo OSI no es muy estrecha. La principal diferencia es que el IP de TCP/IP puede interactuar con diferentes manejadores de dispositivos simultáneamente. Esta característica es la que permite la interconexión de redes: un sistema que está conectado a varias redes puede enviar paquetes de una red a otra sin problema alguno.

Dado el gran crecimiento en la base instalada que ha tenido TCP/IP en los últimos años y que gran parte de los desarrolladores de sistemas operativos lo incluyen, se ha convertido en el protocolo de red por excelencia para los sistemas abiertos (sistemas, *hardware* y *software*, que estandarizan arquitectura y comunicación para la interconectividad).



#### 4. INTERCONECTIVIDAD

En capítulos anteriores se han tocado temas relacionados con la comunicación de datos de una manera superficial. Ahora, se hará un análisis más detallado de este tema, considerando desde principios básicos hasta tecnología de punta para la implantación de redes de comunicación de datos.

En años recientes se ha manejado que la década de los 90's marca la era de las comunicaciones, al menos es lo que se intenta. Las empresas actuales buscan mantener flujo continuo de información mediante la comunicación, o *interconectividad*, entre los sistemas de cómputo de las oficinas centrales y sus filiales para compartir información y recursos. Este es otro factor importante a considerar cuando se hace el análisis para la elección de un sistema de cómputo centralizado.

Los sistemas multiusuarios basados en UNIX pueden comunicarse entre sí, ya que UNIX fue creado pensando en esta necesidad. Las redes de área local basadas en NetWare también lo hacen a corta distancia, pero cuando se requieren interconectar en forma remota lo hacen de una manera menos eficiente y, en muchos casos, muy costosa.

Dependiendo de la naturaleza de la comunicación que se requiera, ya sea bajo red o líneas seriales se deben de tomar en cuenta un determinado número de protocolos y estándares.

Para los sistemas basados en UNIX la comunicación serial asíncrona, como se mencionó anteriormente, está interconstruida en el sistema operativo y es relativamente fácil implantarla. También existen módulos adicionales (que son una combinación de *hardware* y *software*) que pueden ser implantados para obtener comunicación serial síncrona y de red.

Las redes de área local basadas en NetWare no cuentan con estas facilidades, sin embargo, también existen módulos opcionales similares a los de UNIX para tener resultados confiables.

#### **4.1 Conceptos básicos de comunicaciones**

Para empezar, se debe tomar en cuenta que para que pueda existir comunicación se requiere de tres elementos principales: *un transmisor* (quien envía el mensaje), *un medio de comunicación* (por el que viaja el mensaje) y *un receptor* (quien recibe el mensaje). Para la comunicación de datos, los elementos mínimos son también tres: Un DTE (*Data Terminal Equipment*) que es cualquier dispositivo digital, ya sea una terminal, una impresora o una computadora que transmite y/o envía datos; un DCE (*Data Communication Equipment*) que es cualquier otro dispositivo conectado a la línea que manipula las señales o datos transmitidos; y un *medio de comunicación* por el cual son transmitidas las señales, frecuentemente se utilizan las líneas telefónicas.

Los DTEs son *digitales* y operan en sistema binario, pero las líneas telefónicas transmiten en forma analógica. Para lograr que el transmisor y el receptor puedan comunicarse con éxito a través de una línea telefónica analógica, es necesario un DCE que *module* y *demodule* las señales de digital a analógico y viceversa. Este dispositivo se llama *modem* y se muestra en la Figura 4-1.

Hoy en día existen servicios de transmisión digital de datos, pero se utilizan aún las líneas analógicas convencionales porque la red telefónica tiene una enorme cobertura por las décadas que ha llevado su expansión. Ya sea que la transmisión de datos se realice en líneas analógicas o digitales, los costos por estos medios de transmisión pueden llegar a ser determinantes en el presupuesto para la comunicación de datos.

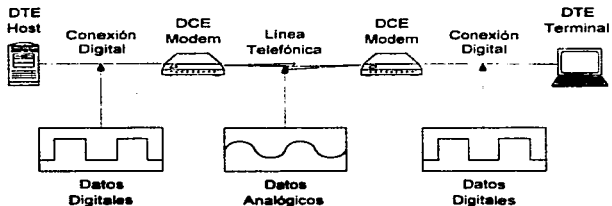


Figura 4-1 Función de un módem

Otra palabra que se debe siempre de recordar en la comunicación de datos es *protocolo*, que define procedimientos necesarios para iniciar y mantener una comunicación, como ya se ha visto en capítulos anteriores.

Uno de los aspectos que controlan los protocolos es la dirección en el flujo de datos sobre la línea telefónica. En este sentido existen tres métodos:

- *Simplex*. La transmisión fluye en una sola dirección.

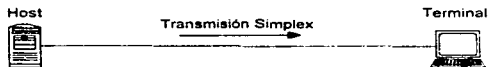


Figura 4-2 Transmisión Simplex

- *Half-Duplex*. La transmisión se realiza en dos direcciones, la única restricción es que sólo se puede hacer en un sentido a la vez.

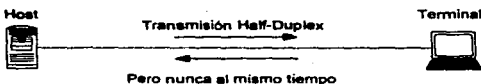


Figura 4-3 Transmisión Half-Duplex

- **Full-Duplex.** La transmisión se realiza en dos direcciones y puede ser simultáneamente.

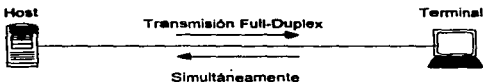


Figura 4-4 Transmisión Full-Duplex

Por último, un concepto por el cual varios organismos internacionales involucrados en el ambiente de comunicación de datos han trabajado es por el establecimiento de *estándares de comunicación*. Este concepto permitiría al usuario conectar productos de diferentes fabricantes de manera transparente. Desafortunadamente, hoy en día no se ha podido realizar este sueño al 100%, ya que los fabricantes siempre desean diferenciar sus productos de los de la competencia al incluir características diferentes o adicionales a las que marcan los estándares. A pesar de estos inconvenientes, la estandarización se está haciendo más sólida a medida que los usuarios la requieren y la exigen a sus proveedores.

## 4.2 Métodos de transmisión y protocolos comunes

Existen dos métodos de transmisión serial de datos que han cubierto la mayoría de los requerimientos de los usuarios. Dependiendo de la aplicación planeada, cada método ofrece ventajas y desventajas al permitir diferentes velocidades de transmisión de datos a diferentes costos.

### 4.2.1 Comunicaciones asíncronas

Cuando un usuario teclea datos en su terminal para que sean recibidos por el *host* a través de una línea serial, éste no puede tener la precisión de teclear a una velocidad constante y el *host* recibe los caracteres teclados en intervalos de tiempo diferentes. El *host* no tiene forma de saber cuándo se recibirá el siguiente carácter.

Debido a este problema de *continuidad* en el flujo de datos, es necesario agregar bits extra antes y después de cada carácter para indicar al dispositivo receptor el inicio y término de cada carácter para poder establecer sincronía entre ambos. Estos bits se conocen como bit de arranque (*start bit*) y bit de parada (*stop bit*). Adicionalmente, en ocasiones se agrega un bit más, conocido como bit de paridad (*parity bit*) para detectar errores de transmisión. Este método es conocido como comunicación asíncrona. La Figura 4-5 muestra los bits de arranque y de parada en la transmisión de un carácter:

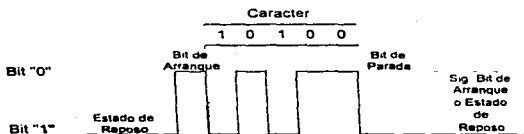


Figura 4-5 Comunicación asíncrona

La transmisión asíncrona es normalmente utilizada en ambientes de captura en terminales tontas (que no tienen capacidad de procesamiento) debido a que no se requieren altas velocidades de transmisión. Las principales ventajas que ofrece este tipo de transmisión incluyen la selección variable de velocidades de transmisión y el rápido establecimiento de la sincronía entre los dispositivos. Adicionalmente, requiere de dispositivos de costo moderado, lo que le permite ser una solución de comunicación de datos viable para aplicaciones de baja velocidad. Sin embargo, los tiempos "muertos" de transmisión entre carácter y carácter hacen ineficiente el aprovechamiento de la línea de comunicación. Las terminales tontas no tienen una memoria temporal (*buffer*) de transmisión por lo que, si existe un error de transmisión de un dato, éste debe ser teclado otra vez hasta que se reciba con éxito.

#### 4.2.2 Comunicaciones síncronas

No es un secreto que el tamaño, complejidad y configuración de las redes privadas de comunicación han crecido rápidamente. Para implantar complejas redes de información es necesario contar con protocolos de comunicación síncrona de alta velocidad y gran confiabilidad en la transmisión de datos.

A diferencia de la transmisión asíncrona, la transmisión síncrona utiliza sofisticados modems para sincronizar el transmisor con el receptor mediante señales de reloj. Se transmiten grandes bloques de datos (*frames*) a la vez y esto permite aprovechar eficientemente la línea de transmisión. La Figura 4-6 muestra un *frame* de SDLC (será visto en breve) en la transmisión de datos:

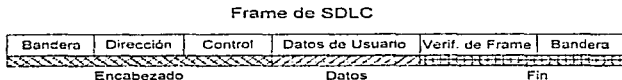


Figura 4-6 Comunicación Síncrona

Debido a la adición de características sofisticadas en la transmisión, como algoritmos de sincronización por reloj, el modo síncrono es más caro de implantar que el modo asíncrono. Sin embargo, a pesar de los costos de los dispositivos, este método ofrece grandes ventajas como la transmisión casi libre de errores, gran velocidad de transmisión y, por consecuencia, reducción de costos de línea de transmisión. A continuación se mencionan los protocolos de comunicación síncrona más comunes.

**SDLC** (*Synchronous Data Link Control*). Es un protocolo de comunicación Full o Half-Duplex orientado a bits. Ha reemplazado al *BISYNC* (*Binary Synchronous Communication Protocol*) de los equipos 3270 para convertirse en el de mayor uso como protocolo síncrono. Ha sido utilizado por IBM en sus implantaciones de redes SNA (*System Network Architecture*).

**HDLC** (*High-Level Data Link Control*). Es protocolo estándar internacional de la CCITT. Opera a través de un amplio rango de ambientes de red de comunicación al igual que en líneas Full-Duplex, reduciendo el tiempo de transmisión.

### **4.3 Redes de área amplia (WAN)**

Cuando los requerimientos de comunicación van mas allá de conexiones ocasionales con baja demanda en la transmisión de información, es necesario considerar tecnologías de comunicación que nos permitan soportar estas demandas así como requerimientos para compartir recursos. Las redes de área amplia (**WAN**, *Wide Area Network*) permiten realizar intercambio de información en sitios distantes, ya sea a través del país o a nivel internacional, mediante enlaces remotos con líneas de comunicación de alta velocidad.

#### **4.3.1 Conmutación de paquetes**

La conmutación de paquetes es una técnica de comunicaciones de área extendida que utiliza una red de datos compartida en lugar de líneas dedicadas. Los datos del usuario se subdividen en cortos paquetes que se transmiten por separado a través de la red. Una vez llegados al destino, los paquetes son vueltos a ensamblar en su formato original. El enrutado de los paquetes se realiza de una de dos maneras: por *datagrama* o por *circuito virtual*.

En datagrama cada paquete es tratado en forma independiente, de modo que varios paquetes pertenecientes a la misma llamada pueden recorrer rutas distintas. En los circuitos virtuales se establece una ruta específica para cada llamada; todos los paquetes de la llamada siguen la misma ruta a través de la red. La mayoría de las redes actuales utilizan el método de circuito virtual.

Algunas ventajas de la conmutación de paquetes son:

- Mayor rendimiento de la línea de comunicación, ya que los enlaces de larga distancia son compartidos dinámicamente por numerosas llamadas.
- Conversión de velocidad de datos, los usuarios que transmiten a diferentes velocidades pueden intercambiar información.
- Manejo de la carga de trabajo.
- La red brinda un almacenamiento temporal (*buffering*) para soportar incrementos de transmisión sin que se produzca bloqueo.
- Estandarización de acceso a la red, permitiendo la implantación de equipos de diferentes fabricantes en forma transparente.

#### **4.3.2 Conexiones virtuales**

La conmutación de paquetes se diferencia de las conexiones punto a punto (líneas de comunicación dedicadas, como lo que hasta ahora se ha visto) en que utiliza *circuitos virtuales*. En otras palabras, realiza conexiones *lógicas* ya



que no hay liga permanente entre dos puntos que intercambian datos. Cada paquete de datos lleva consigo la dirección del destinatario por lo que no importa qué ruta tome, siempre llegará al mismo lugar (como lo muestra la Figura 4-7). Existen servicios de conmutación de paquetes que permiten el acceso a cualquier otro punto. Es similar a las redes telefónicas: para llamar a un amigo se debe saber su número de teléfono (dirección), no se sabe a través de qué central(es) telefónica(s) es canalizada la llamada pero siempre se puede establecer la comunicación.

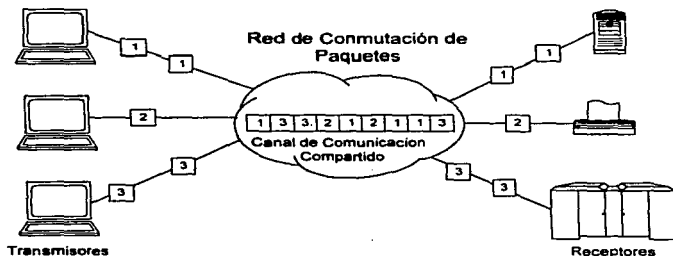


Figura 4-7 Conmutación de paquetes

Las redes de conmutación de paquetes pueden definirse como redes interactivas de tiempo real, ya que ofrecen el uso eficiente de circuitos de comunicación. Su principal ambiente de aplicación es la de interconexión entre usuarios geográficamente separados. Las redes de conmutación de paquetes proveen la flexibilidad requerida para aplicaciones de tiempo real; también ofrecen corrección de errores de transmisión y soportan perfectamente grandes volúmenes de información.

Existen tres tipos de redes de conmutación de paquetes: *públicas*, *privadas* y una combinación de ambas llamadas *híbridas*. Todas ellas reducen el tiempo de transmisión de datos y por consecuencia, los costos, al utilizar la misma línea de comunicación para varias aplicaciones simultáneamente sin importar la distancia. Las grandes corporaciones crean y mantienen redes privadas, mientras que las empresas medianas utilizan normalmente servicios públicos.

#### **4.3.3 Componentes básicos**

Son cinco los componentes para poder realizar una transmisión de datos por conmutación de paquetes.

1. LAC: *Local Access Components*, componentes de acceso local.
2. PAD: *Packet Assemblers/Disassemblers*, ensamblador/desensamblador de paquetes.
3. PN: *Packet Switching Nodes*, nodos de conmutación de paquetes.
4. NL: *Network Links*, líneas de conexión.
5. NMS: *Network Management System*, sistema de administración de red.

Los componentes de acceso local como terminales, impresoras y modems mueven los datos hacia el PAD mas cercano. El PAD provee acceso a la red y asegura la compatibilidad entre los dispositivos de los usuarios y la red de conmutación de paquetes. En todos los casos la información debe tener el formato correcto antes de llegar al nodo de conmutación de paquetes. El PAD "empaca" los datos de la terminal y los envía al nodo para que sean ruteados. Adicionalmente a "empaca", el PAD se encarga de "desempacar" cuando llegan datos antes de que sean recibidos por la terminal de destino. Otras funciones del PAD incluyen: establecimiento de llamadas, conversión de protocolos, conversión de códigos y emulación de protocolos.

El nodo rutea cada paquete al destinatario adecuado. También se encarga de realizar diagnósticos de la red y establece las conexiones entre diferentes redes.

La Figura 4-8 muestra estos componentes.

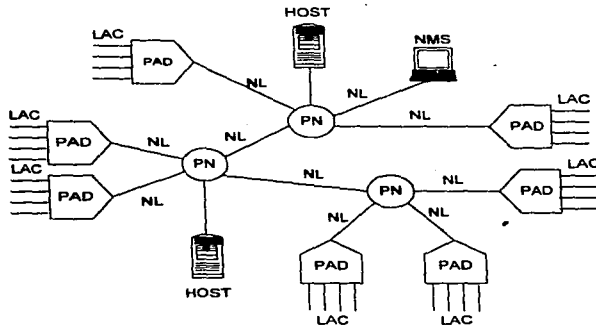


Figura 4-8 Componentes de una red por conmutación de paquetes

Las líneas de conexión pueden ser líneas telefónicas, microondas, sistemas satelitales, entre otras. Estas líneas son la liga primordial de la red y son llamadas *backbone* (espina dorsal) de la red.

La función básicas del sistema de administración de la red es mantener la base de datos de la red. Aquí se almacenan una copia maestra de la tabla de ruteo, las configuraciones de cada nodo y el perfil de cada interface del usuario. En caso de una contingencia, este sistema es capaz de transmitir esta información a cada punto con el fin de reestablecer la conectividad.

Adicionalmente, este sistema puede monitorear la operación de la red para detectar fallas y accesos no autorizados.

#### **4.3.4 X.25**

X.25 es un protocolo de transmisión de datos, creado en 1976 por la CCITT, que utiliza el método de conmutación de datos. Su operación de basa en los tres primeros niveles del modelo OSI (ver el Capítulo anterior para mayores detalles del modelo).

Existen básicamente tres puntos que, al implantar X.25, son directamente mejorados:

1. *Dispersión de usuarios.* Entre más dispersos se encuentran los usuarios, más económico resulta implantar la conmutación de paquetes, ya que reduce el número de líneas de comunicación. Las redes de conmutación de paquetes comparten el ancho de banda entre los usuarios y, por consecuencia, reduce el costo de transmisión por usuario.
2. *La necesidad de conectividad.* Entre más dispositivos se puedan alcanzar desde una terminal, mayor conectividad tendrá esa terminal; entre mayor conectividad se tenga, mayor será el ahorro en equipo. La comunicación punto a punto limita la conectividad ya que provee acceso a un solo recurso. No todas las aplicaciones requieren de acceso a un gran número de recursos pero, en el caso de requerirlo X.25 puede proporcionar ese beneficio.
3. *Patrones de tráfico.* Cuando se subutiliza una línea de comunicación se están desperdiciando recursos. X.25 permite el acceso simultáneo de varias aplicaciones utilizando al máximo el ancho de banda de transmisión.

Algunas de las aplicaciones típicas que incluyen X.25 son: correo electrónico, conexión de cajeros automáticos, aprobaciones de tarjetas de crédito, control de inventario y reservaciones en línea, por mencionar algunas.

#### 4.3.5 *Frame Relay*

Este es un protocolo de redes de comunicación diseñado para manejar intensas aplicaciones de transmisión de datos. Está basado en las siguientes cuatro características:

- Gran velocidad de transmisión de datos.
- Poco retraso en la red.
- Gran conectividad.
- Optimización del ancho de banda.

*Frame Relay* es una mejora a la transmisión por conmutación de paquetes. Está diseñado específicamente para manejar problemas en la diversidad de tamaños, patrones y volúmenes de paquetes.

Las redes de área local (LAN) interconectan a las estaciones de trabajo en un espacio determinado como edificios y campus universitarios. Pero la interconexión de varias LANs a través de diferentes localidades requiere también del uso de los servicios de redes de área amplia (WAN) y entonces comienzan los retos. Primero, los usuarios que han utilizado los servicios de una LAN están acostumbrados a la rapidez en el tiempo de respuesta y esperan obtener lo mismo de una WAN. Segundo, el ancho de banda de una WAN es más caro que el de una LAN. Por último, el perfil del tipo de datos que son transmitidos, desde la transferencia de archivos enormes hasta complicadas imágenes y vídeo. Este tipo de aplicaciones generan gran tráfico en la red y deben ser soportadas por dispositivos capaces de transmitir cualquier tipo de datos a altas velocidades, además de optimizar el ancho de banda y no tener retrasos en la transmisión. *Frame Relay* es una interface de red diseñada para enfrentar estos requerimientos.

*Frame Relay* es, de hecho, un protocolo de comunicaciones. Utiliza parte del nivel 2 (*data link layer*) y todo el nivel 1 (*physical layer*) del modelo OSI al

realizar tareas de establecimiento de conexiones y transferencia de datos a través de la conexión. A diferencia de X.25 (que utiliza los tres primeros niveles del modelo OSI), *Frame Relay* deja el trabajo de corrección de errores de transmisión a dispositivos inteligentes que están situados en los extremos de la red. Esto reduce el procesamiento requerido por los nodos de red y, en consecuencia, reduce el tráfico de datos y el retraso en la llegada de información de un lado a otro.

Existen muchos beneficios al utilizar una red de conmutación de datos para interconectar LAN's a través de *Frame Relay*. A principios de los 80's, cuando la conmutación de datos comenzaba, el número de LANs era mínimo. Con esta baja demanda, la tecnología para transmitir grandes volúmenes de datos no requería de gran sofisticación. Pronto los requerimientos de productividad al implantar procesamiento distribuido se incrementó y así lo hizo también el número de LANs. El desarrollo de *Frame Relay* fue conceptualizado para cubrir estas necesidades de interconectar LANs a través de puentes y ruteadores. Los principales beneficios al utilizar *Frame Relay* son:

- Las LANs obtienen acceso rápido a las WANs.
- Mediante los métodos de compresión de datos y optimización del ancho de banda de *Frame Relay*, existe una gran disponibilidad de tiempo de transmisión.
- Gran cantidad de puentes y ruteadores de LAN utilizan un *backbone* común.
- Las LANs pueden ser controladas y administradas desde un punto de vista global.
- Se pueden implementar de mejor forma aplicaciones transaccionales, cliente/servidor, CAD/CAM, manipulación de gráficos, transmisión de faxes, intercambio electrónico de datos, y multimedia.

#### 4.4 Ejemplo de implantación

Ya se han definido las partes fundamentales del funcionamiento de una LAN y una WAN. Ahora se planteará a manera de ejemplo un caso hipotético en el cual se forme una combinación de redes de área local, sistemas multiusuarios y una red de área amplia.

##### Perfil de la empresa:

Una empresa transnacional de fabricación de motores tiene sus oficinas principales y una planta de producción en San Francisco, oficina de diseño en México, oficinas administrativas en Detroit, oficina de ventas internacionales en New York y oficinas de ingeniería en París.

##### Requerimiento:

Se va a iniciar un proyecto para el desarrollo de un nuevo motor. Los diseñadores en México deben intercambiar los diseños diariamente con el departamento de ingeniería en París para la validación de especificaciones del motor. Estos diseños son desarrollados en CAD/CAM y, al ser información de formato gráfico, se requiere de una gran velocidad en la transferencia de datos. El departamento de ingeniería consulta una vez a la semana información de las especificaciones del producto con la planta de producción mediante una aplicación que requiere de acceso en línea, por lo que debe existir siempre una línea de comunicación abierta. Una vez terminado el proyecto debe enviarse toda la información referente al motor a la planta de producción para la fabricación.

##### Implantación:

La Figura 4-9 muestra los componentes necesarios para la implantación de la solución.

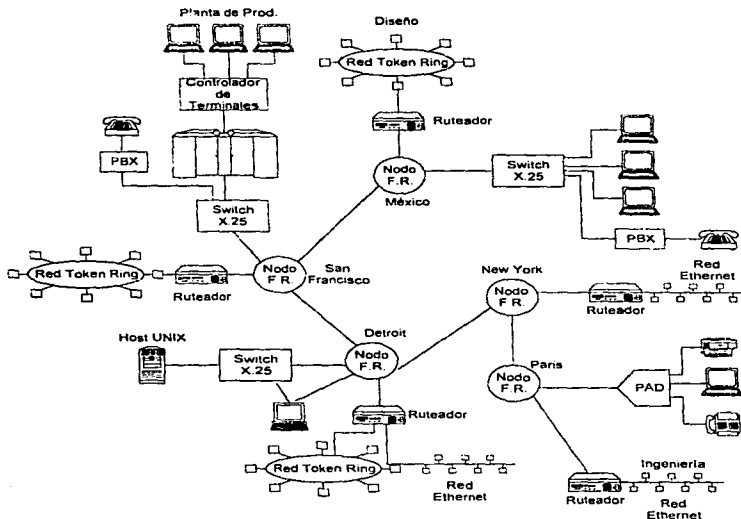


Figura 4-9 Implantación de interconectividad compleja

El departamento de diseño en México cuenta con una red de área local bajo el sistema operativo de red Netware y topología Token Ring. El diseño se hace por separado de acuerdo a la especialidad de cada diseñador. Para tener acceso a la red de área amplia es necesario instalar un ruteador que sirva de enlace entre la red Token Ring y los nodos de comunicación *Frame Relay*. Esto permitirá una transferencia de datos en alta velocidad y sin errores. Debe



configurarse el ruteador de manera que tenga la dirección de red del departamento de ingeniería en París para que se asegure que la información llegue al lugar correcto.

El departamento de ingeniería en París cuenta con una red de área local bajo sistema operativo de red Netware y topología *Ethernet*. También es necesario instalar un ruteador con interfaces *Ethernet* y *Frame Relay* para tener acceso a la red de área amplia.

Las oficinas centrales cuentan con un mainframe con acceso directo a la red de área amplia. En este sistema se encuentra concentrada toda la información relativa a la operación completa de la empresa.

Las oficinas de Detroit y New York sirven como enlace de paso entre la comunicación de México y París. Sin embargo, éstas se pueden enlazar a cualquier punto de la red sin importar el sistema que esten utilizando.

Adicionalmente, existe comunicación telefónica a través de la red digital de datos entre las oficinas de México y San Francisco mediante el acceso directo de los conmutadores (PBX) a la red de área amplia.

Este tipo de instalaciones eran imposibles de implantar en el pasado debido a que no existían los dispositivos de interconectividad que pudieran servir de enlace entre los diferentes ambientes, interfaces y protocolos de comunicación, pero gracias a la tecnología actual y la estandarización de los protocolos de comunicación ahora es posible tener varias alternativas de conectividad en el que ni el fabricante del equipo de comunicaciones ni el de los sistemas de cómputo sean factores determinantes.

## 5. CONSIDERACIONES EN LA TOMA DE DECISIONES

Existen gran cantidad de aplicaciones comerciales y herramientas de desarrollo disponibles en el mercado tanto para sistemas multiusuario como para redes de área local.

Dado que los sistemas operativos de red y multiusuario son de naturaleza muy diferente, las aplicaciones y herramientas deben ser desarrolladas para que se ejecuten de manera óptima en cada uno de los ambientes, pero no indistintamente. Por ejemplo, el procesador de palabra con mayor aceptación en el mercado mundial, WordPerfect, se ha desarrollado en versión MS-DOS, red, UNIX, MVS y otras. Cada una de las versiones presenta las mismas características operacionales para el usuario e incluso los documentos pueden ser utilizados en cualquier versión pero, en el fondo, son totalmente diferentes. A este tipo de aplicaciones se le conoce como multiplataformas y existe una versión diferente para cada ambiente. De hecho, algunas de las aplicaciones de gestión administrativa, varias de ellas mexicanas, se han desarrollado en igual o mayor número de versiones del ejemplo anterior. Esto responde a la necesidad de compartir aplicaciones y datos en diferentes ambientes operativos sin perder la consistencia de la información.

A pesar de los esfuerzos de los fabricantes por mantener la misma versión en las diferentes plataformas, algunos productos no pueden mantener su actualización ya que dependen muchas veces de las versiones correctas del sistema operativo en el cual va a ejecutarse la aplicación.

Ambos ambientes han implantado las interfaces gráficas de una manera similar, pero a la vez muy diferente en cuanto al desarrollo tecnológico. La elección de un ambiente u otro en sentido gráfico requiere de un análisis minucioso de costo-beneficio.

## 5.1 Ambientes gráficos

En la actualidad la mayoría de las aplicaciones incluyen la interface del usuario en modo gráfico, a través de ventanas, iconos, manejo de un dispositivo apuntador (*mouse* o similar) e incluso video y sonido.

A los usuarios les gusta el ambiente gráfico porque es fácil de usar y porque, en interconexión de redes, es posible compartir archivos e impresoras. Sin embargo, a medida que las organizaciones evolucionan hacia ambientes de procesamiento distribuido, los requerimientos de tecnología de *hardware* y *software* para soportarlo crecen proporcionalmente al volumen de información gráfica transmitido.

### 5.1.1 Windows 3.11

Es un hecho que el *ambiente operativo* gráfico Microsoft Windows 3.11 es el líder en este concepto para la plataforma de PCs. Su éxito se basa en el aprovechamiento de la base instalada de MS-DOS como el sistema operativo por excelencia de las computadoras personales basadas en procesadores Intel. El hecho de que Windows 3.11 sea un ambiente operativo y no un sistema operativo se basa en que éste último administra los recursos de la computadora e interactúa directamente con ella y un ambiente operativo no. Es obligatorio instalar primero el sistema operativo MS-DOS y después Windows 3.11 para obtener la funcionalidad gráfica, como se muestra en la Figura 5-1. Las limitaciones de operación de MS-DOS, como la administración de memoria (que sólo puede administrar hasta 1Mb), deben de ser solucionadas mediante manejadores externos (*drivers*) para que Windows 3.11 pueda tener acceso a los recursos que MS-DOS no puede controlar.

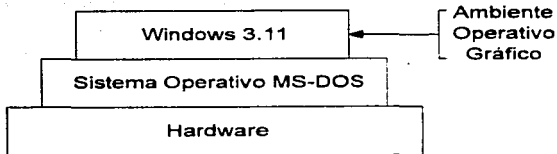


Figura 5-1 Esquema general de Windows 3.11

Aún con estas limitantes, Windows 3.11 se encuentra instalado y funcionando en millones de computadoras a nivel mundial. Los desarrolladores de *software* comercial ven en este mercado un potencial enorme de crecimiento y crean sus aplicaciones para ser compatibles con este mercado.

Los productos de *hardware* requieren de manejadores para poder ser compatibles con Windows 3.11. Para los productos más comerciales los manejadores se encuentran incluidos en Windows 3.11; si se requiere instalar un producto de *hardware* que no tiene el manejador incluido en Windows 3.11, el fabricante debe proporcionarlo. Desafortunadamente un manejador adicional no en todos los casos funciona correctamente ya que depende de la configuración inicial tanto de MS-DOS como de Windows 3.11.

En la actualidad existen cientos de productos, entre *hardware* y *software*, compatibles con Windows 3.11 que permiten aumentar su funcionalidad y conectividad a servidores de red. Estos incluyen impresoras, monitores, tarjetas de red, lectores de CD-ROM, tarjetas de sonido, entre otros.

Una extensión del Windows 3.11 es *Windows para Trabajo en Grupo*. Esta versión tiene, además de las características de Windows 3.11, capacidad de conectarse a una red pequeña con un protocolo de comunicación sencillo. De

esta manera se puede crear una red de área local para pequeños grupos de trabajo sin la necesidad de contar con un costoso servidor de archivos dedicado, ya que cada estación de trabajo puede compartir sus recursos con las demás.

### **5.1.2 Windows 95**

Como una necesidad del mercado de contar con un sistema operativo con interface gráfica como estándar, Microsoft lanzó Windows 95. Este sistema operativo, operacionalmente, es mucho mas estable que Windows 3.11. Tiene incluidos cientos de manejadores para *hardware* adicional con la intención de que el usuario solo instale y use (*Plug-and-Play*), eliminando el riesgo de establecer una mala configuración de las opciones. La administración de recursos es muy buena ya que puede controlar decenas de Megabytes de memoria directamente sin la necesidad de manejadores externos. Tiene como estándar la facilidad de conexión a redes de área local, conexión a Internet y el soporte a la mayoría de las aplicaciones compatibles con Windows 3.11.

Su principal desventaja es que requiere de muchos recursos de *hardware* como gran cantidad de memoria, amplio espacio en disco duro y un procesador 486 o superior para que pueda correr con un buen rendimiento.

A pesar de que su conceptualización en el diseño es excelente, los usuarios todavía están renuentes a utilizarlo porque en las primeras versiones existieron fallas operacionales y Microsoft postergó el lanzamiento varias veces antes de tener lista la versión comercial definitiva.

### **5.1.3 X Windows y OSF/Motif**

Para el sistema operativo UNIX, X Windows y OSF/Motif son los componentes más populares como manejadores de ambiente gráfico. No se encuentran

incluidos en el sistema operativo básico ya que ambos son adiciones a UNIX para la administración del ambiente gráfico. Su estructura funcional se muestra en la Figura 5-2:

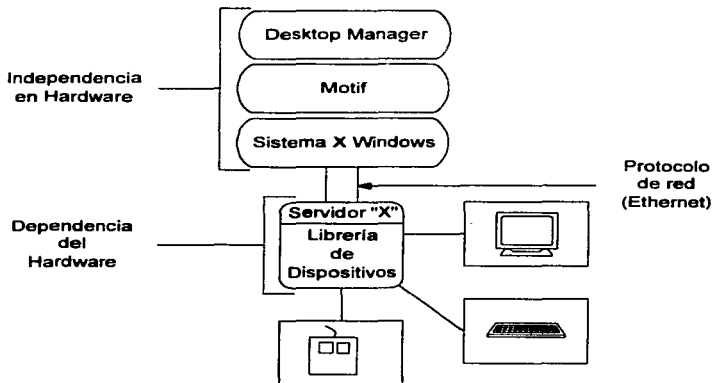


Figura 5-2 Esquema general de X Windows y OSF Motif

El Sistema X Windows interactúa con UNIX para obtener los recursos necesarios para presentar el ambiente gráfico para el usuario. Dado que UNIX es un sistema operativo que administra sus recursos muy eficientemente, su comunicación con X Windows es estable y fluida.

Las funciones principales del Sistema X Windows son las siguientes:

- Administración del despliegado de las gráficas de bajo nivel, como el teclado, el mouse y las gráficas en pantalla.

- Proveer librerías que contienen las herramientas de programación para definir ventanas y aplicaciones X.

Para el manejo y definición de los elementos de interacción con el usuario en modo gráfico existe OSF/Motif. Sus principales funciones son las siguientes:

- Control general de ventanas.
- Administración de los atributos de las ventanas como funcionalidad, apariencia, tamaño, color y títulos.
- Definición de iconos en cuanto a tamaño, forma y títulos.
- Control de los botones del *mouse* y teclado.
- Definición de opciones de menú.

La mayoría de las funciones que el usuario puede realizar en UNIX pueden hacerse mediante la interface gráfica. Adicionalmente, puede controlar aplicaciones tanto en el sistema en donde se esté ejecutando como en sistemas remotos, si existe alguna conexión entre ellos.

La implantación de interfaces gráficas en UNIX no es sencilla ya que se requiere de amplios conocimientos de UNIX y conectividad en red, sin embargo su rendimiento es excelente cuando se ha realizado con éxito.

Una desventaja es que, como UNIX utiliza grandes recursos del sistema, resulta costoso implantar este ambiente gráfico para cada usuario del sistema global. De hecho, resulta mas económico utilizar *software* emulador de X Windows en estaciones de trabajo con Windows 3.11 que hacerlo directamente en UNIX.

## 5.2 Bases de datos relacionales

Uno de los principales factores para que una aplicación de procesamiento de datos tenga un rendimiento adecuado a las demandas, es un *Sistema de Bases de Datos Relacional* (DBMS) confiable y poderoso. Se implanta normalmente en ambientes Cliente/Servidor, el DBMS reside en el Servidor y contiene herramientas para programación en el Cliente (estaciones de trabajo en donde se ejecuta la aplicación).

Para empezar, una base de datos se define como una tabla que consiste en renglones y columnas. Un renglón corresponde a cada registro de datos y una columna corresponde a un campo del registro.

El que sea *relacional* significa que se pueden establecer ligas entre los diferentes campos y registros de una o varias bases de datos para obtener un resultado específico. De ésta manera es posible mantener actualizada una base de datos de Cuentas por Pagar a partir de los datos que se encuentran en la base de datos de Clientes.

Un DBMS es el "motor" de aplicación genérica para la administración de bases de datos de alto volumen de información. Este no es una aplicación comercial, sino que provee de todas las herramientas para crear una a la medida de los requerimientos. Contiene un lenguaje propio de programación avanzado con rutinas que permiten la administración eficiente de gran cantidad de datos, la seguridad, los lineamientos principales para transacciones, controles de concurrencia y otras rutinas que, de otra manera, deberían programarse por separado. Normalmente llevan implantado SQL (*Sequenced Query Language*), un lenguaje de programación de alto nivel que es compatible con muchas aplicaciones existentes.



Muchos de los DBMS han sido originalmente productos del ambiente **multiusuario**, pero con el incremento en la participación de mercado de las microcomputadoras, éstos han sido portados hacia sistemas operativos para equipos "pequeños". Sin embargo, los DBMS mas comerciales tienen un desempeño óptimo en el ambiente para el cual fueron creados: UNIX.

La principal ventaja en utilizar un DBMS es la integración que se obtiene en el desarrollo de las aplicaciones, sin importar en qué plataforma de *hardware* se desarrolle. Es decir, se puede desarrollar una aplicación en un sistema multiusuario UNIX y después implantarla en una red NetWare sin modificaciones. Esto da la posibilidad de crear aplicaciones en diferentes ambientes de acuerdo a los requerimientos.

### **5.2.1 Características de un DBMS**

Todos los DBMS comerciales tienen características propias, pero en general deben cumplir al menos con las siguientes:

- *Soporte a ejecución de multitareas.* Permite ejecutar procesos simultáneos.
- *Distribución de la bases de datos.* Permite expandir los datos a través de los volúmenes del disco o, aún mejor, a través de varios servidores de bases de datos separados geográficamente.
- *Control de transacciones.* Permite llevar registro de todas las transacciones de la aplicación para consultas o deshacer una actualización de registro.
- *Control de concurrencia.* Permite el bloqueo de registros y archivos para ambientes multiusuarios.
- *Seguridad de los datos.* Brinda acceso a los usuarios que tienen los permisos adecuados para ciertos procesos. Esta seguridad debe ser igual o superior que el acceso que provee el sistema operativo.
- *Soporte para múltiples protocolos de red.* Permite la implantación de ambientes Cliente/Servidor para procesamiento distribuido. Este es uno de

los puntos más importantes de un DBMS, ya que explota la interconectividad existente en muchas instalaciones corporativas.

### **5.2.2 Soporte y herramientas para el sistema cliente**

Las herramientas de programación para el sistema Cliente deben proporcionar las siguientes funcionalidades:

- *Interface de programas de aplicación.* Herramientas para la automatización de cualquier proceso en el que se requiere el acceso a datos del Servidor.
- *Soporte de varios lenguaje de programación.* Permite crear rutinas que puedan ser llamadas desde otras aplicaciones hechas en otro lenguaje de programación.
- *Soporte de 4GL.* Los lenguajes de cuarta generación (4GL) son lenguajes de alto nivel superiores a C o Pascal. El 4GL ha sido optimizado para desarrollar aplicaciones de administración de datos ya que contienen comandos que permiten explotar eficientemente la información contenida en las bases de datos relacionales.
- *Herramientas administrativas.* Permiten la creación y modificación de bases de datos, administrar usuarios, y otras funciones desde una estación de trabajo.
- *Herramientas interactivas.* Permite el acceso directo a las bases de datos para la consulta y explotación de la información sin estar en una aplicación específica.

### **5.2.3 Ejemplos de DBMS**

Los plataformas de DBMS mas comerciales han tenido su origen en sistemas multiusuario centralizados y en la actualidad se han actualizado para cubrir los requerimientos del mercado en ambientes Cliente/Servidor. Cada uno tiene

características propias de operación, pero cubren al menos con los requerimientos anteriores.

### **Oracle**

El Oracle Server es el DBMS comercial más popular en la actualidad y está disponible para una gran variedad de plataformas de *hardware*. Fue originalmente desarrollado como la base de datos relacional para VAX MVS en ambientes de red.

### **Informix**

El Informix On-Line es un producto desarrollado inicialmente para plataformas UNIX. Tiene gran soporte para este sistema operativo y toma ventaja de su arquitectura interna. El desarrollo para otras plataformas no ha sido sólido pero existen versiones para NetWare y otros. El Informix 4GL es un lenguaje de programación muy poderoso gracias a la retroalimentación que se obtiene de los programadores.

### **Sybase**

Aunque es el menos popular de los tres, el Sybase SQL Server es el producto con mejor desarrollo técnico, ya que los desarrolladores de Sybase trabajan junto con los fabricantes de sistemas operativos para obtener un producto altamente estable en diferentes ambientes. El soporte de lenguajes de programación de los sistemas Clientes es excelente porque permite la integración de módulos de Sybase con programas de aplicación comercial como Microsoft Excel, entre otros.

## **5.3 Implantaciones híbridas**

Ya se han tocado los temas de sistemas operativos de red y multiusuario, comunicación e interconectividad, bases de datos e interfaces gráficas. En las

tecnologías *hardware* y *software* de hace una década estos temas eran conceptualizados por separado y se creaban productos que cumplieran con esas tendencias. En la actualidad, es imprescindible hablar de ellos en forma conjunta ya que se requieren herramientas que proporcionen información y forma de operación globalizadas. No es extraño ver ahora sistemas de diferentes fabricantes interconectados y trabajando en forma coordinada para un fin común. La apertura hacia los Sistemas Abiertos y estándares de comunicación han roto las barreras que frenaban el desarrollo tecnológico del procesamiento distribuido y redes cooperativas.

Hay que reconocer que no todos los ambientes de operación, red o multiusuario, cumplen con todas las expectativas de funcionalidad que el mercado demanda. Para poder cubrir estos requerimientos se realiza la integración de varias tecnologías para tomar ventaja de las principales cualidades de cada una. Es decir, se forman *implantaciones híbridas*. La Figura 5-3 muestra el esquema clásico de una implantación de esta naturaleza.

En este esquema, el *host UNIX* es el responsable del servicio de bases de datos (DBMS) y de ejecutar las aplicaciones de las terminales tontas conectadas a través de puertos seriales, los servidores NetWare proveen las aplicaciones de ambiente gráfico para las estaciones de trabajo al igual que los servicios de impresión, el *ruteador* establece la comunicación entre ambas topologías de red bajo el protocolo TCP/IP y la salida hacia una red de área amplia en X.25.

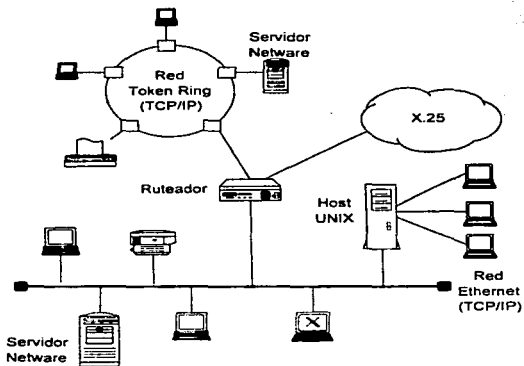


Figura 5-3 Esquema general de implantaciones híbridas

La ventaja de diseñar una configuración como la anterior es aprovechar el excelente rendimiento que ofrece UNIX para la administración de las bases de datos y la facilidad de implantación de ambiente gráfico en NetWare.

Como se mencionó en el capítulo anterior, el protocolo TCP/IP permite manejar diferentes medios de comunicación, por lo que no es difícil entablar comunicación entre cualquier estación de trabajo conectada a la red y el *host* UNIX a través de un emulador de terminal llamado Telnet.

### 5.3.1 SAP, ejemplo de implantación híbrida corporativa

SAP es un sistema integral de aplicación corporativa que contempla todos los procesos comerciales y transacciones de una empresa o grupo de empresas mediante módulos, independientes o integrables entre sí. La integración de

los módulos liga todos los procesos a través de datos y funciones e interactúa con el usuario por medio de una interface gráfica, que es uniforme para todas las áreas de aplicación.

La estructura completa de módulos se muestra en la Figura 5-4:

<b>R/3</b> <b>Client/Server</b> <b>ABAP/4</b>			
SD Sales & Distribution	MM Material Mgmt	PP Production Planning	QM Quality Mgmt
PM Plant Maintenance	HR Human Resources	IS Industrial Solutions	WF Workflow
PS Project System	AM Fixed Assets Mgmt	CO Controlling	FI Financial Accounting

Figura 5-4 Estructura general de SAP R/3

Se basa en la arquitectura Cliente/Servidor a tres niveles: presentación, aplicación y bases de datos. Para ejecutar determinadas tareas pueden utilizarse varios servidores interconectados entre sí mediante redes de comunicación sin perder la integración de los datos y procesos del sistema.

El sistema base SAP-R/3 permite utilizar aplicaciones con altos niveles de prestaciones en diferentes plataformas de *hardware* y de sistemas operativos para adaptarlas a los requerimientos específicos del usuario. Cumple con todos los requisitos de procesamiento actuales y futuros al ofrecer la integración perfecta del *software* y la tecnología y disponer de herramientas de comunicación que evitan al máximo las reinversiones.

La Figura 5-5 muestra las plataformas de *hardware* y *software* así como los diferentes sistemas operativos y DBMS con los que puede convivir:






Hardware 	Sistemas UNIX Bull IBM Digital SNI HP_SUN	AT&T DEC Sequent Bull/Zenith HP (Intel) SNI Compaq IBM ...	IBM AS/400
Sistema Operativo 	AIX SNIIX Digital UNIX Solaris HP-UX	Windows NT	OS/400
Base de Datos 	ADABAS D AIX/DB2 Informix Oracle 7	ADABAS D MS SQL Server 6.0 Oracle 7	DB2/400
Presentación Gráfica 	Windows 3 11, Windows 95, Windows NT, OSF/Motif, OS/2 Warp, Macintosh		Windows 3 11, Windows 95, Windows NT, OS/2 Warp
Lenguajes 	ABAP/4, C, C++		

Figura 5-5 Opciones de plataformas de aplicación de SAP

El sistema es completamente parametrizable, permitiendo adecuar la aplicación a cualquier requerimiento de operación presente y futuro. Para poder realizar estas adecuaciones, SAP-R/3 incluye un grupo integrado de herramientas de desarrollo de cuarta generación con las que pueden instalarse aplicaciones cliente/servidor críticas, ampliar las aplicaciones incluidas o modificar los módulos estándar R/3 instalados. Estas herramientas son parte de un paquete llamado ABAP/4 DW, que es la abreviatura de *Advanced Business Application Programming Development Workbench*.

Las características más importantes de ABAP/4 DW son:

- **Acceso estandarizado a las bases de datos.** Los programas se desarrollan independientemente de la base de datos relacional definida mediante comandos de SQL, permitiendo también el acceso directo a las bases de datos.

- **Infraestructura.** Soporta todos los tipos de *hardware*, sistemas operativos, bases de datos e interfaces gráficas de usuario (GUI) conocidas.
- **Comunicación con sistemas y programas externos.** Con ayuda de la *Remote Function Call (RFC)*, los programadores pueden crear interfaces con otros sistemas y programas externos. Con ayuda de *Object Link Embedding (OLE)*, es posible crear ligas con aplicaciones de escritorio como MS Excel o MS Word.
- **Versiones internacionales.** Permite el desarrollo de programas, pantallas y GUI en los idiomas que soporta SAP.
- **Modificación y creación de pantallas.** Es posible crear y modificar pantallas con atributos especiales o iconos.

La implantación de SAP es sumamente compleja y es necesario realizar un proceso de reingeniería con la asistencia de asesores especializados en este sistema para garantizar su correcto funcionamiento. Por este motivo SAP no es un sistema para todas las empresas ya que resulta muy costoso el proceso de implantación. Lo que es importante resaltar es el alto grado de integración y eficiencia que se obtiene de un sistema que toma ventaja de los Sistemas Abiertos y estándares de la industria.

#### **5.4 Puntos importantes de decisión**

Se deben de tomar en cuenta los siguientes puntos para tomar la decisión correcta en la elección de un sistema de cómputo centralizado o distribuido:

- **Facilidad de interacción con el usuario.** El tipo de presentación para el usuario es un factor que puede aumentar o disminuir la productividad. Entre más compleja sea la operación del sistema, mayor será el rechazo del usuario a utilizarlo. Una interface gráfica muchas veces puede parecer más fácil de utilizar que una de tipo caracter; funciona bien para personas que ya



han tenido contacto con computadoras, pero para un usuario que nunca se ha sentado frente de una computadora puede ser de gran impacto el operar un sistema con múltiples ventanas y un dispositivo apuntador que, por mucho que avance la tecnología, requiere de práctica el poder dominar. Es más fácil acostumbrarse a un teclado y a una pantalla sencilla.

- *Tiempo de respuesta requerido.* No todos los procesos requieren de un tiempo de respuesta inmediato para que se pueda decir que tiene buen rendimiento, tal es el caso de los procesos de conciliación de cuentas bancarias. Este proceso puede ser efectuado durante las horas en que no existe tanta demanda de recursos del sistema. Por otro lado, un proceso de punto de venta requiere de la actualización inmediata de las bases de datos para el control de inventarios, por lo que es necesario el acceso al sistema en línea y en tiempo real.
- *Tipo de información que será transmitida.* Entre menor sea el tráfico de información en un sistema, mejor servicio dará a los usuarios. Los sistemas con procesamiento distribuido deben contemplar mecanismos de balanceo de las cargas de trabajo entre los diferentes servidores para evitar al máximo la transmisión de datos innecesarios. Es importante definir exactamente el tipo de datos que serán transportados por la red de comunicaciones. Un archivo gráfico o una imagen generan gran carga de trabajo a la red mientras que los datos de tipo carácter pueden ser transmitidos con gran facilidad.
- *Volumen y complejidad de la información.* La cantidad de información que será controlada y almacenada por el o los sistemas, el número de procesos que se requieren para obtener un resultado y el número de usuarios que accederán el sistema.

- **Seguridad en el acceso.** Debe de contar con restricción de privilegios a los usuarios para mantener la confidencialidad de la información. Un sistema inseguro puede dar acceso a una persona no autorizada y ocasionar una caída del sistema o la pérdida de información así como la contaminación por virus informáticos.
- **Estabilidad de sistema.** Cuando la operación de toda una empresa se basa en la disponibilidad de un sistema de cómputo, es indispensable contar con mecanismos de recuperación del sistema en caso de caídas. Se deben considerar también dispositivos de interconexión de red que tengan modos alternativos de transmisión cuando se pierde una línea de comunicación.
- **Complejidad en la implantación y/o migración.** Es deseable utilizar los mismos recursos informáticos con los que se cuenta actualmente, como computadoras, sistemas operativos o aplicaciones, sin embargo se debe considerar la rápida obsoletización de los sistemas en la actualidad. Tratar de forzar una implemetación con recursos obsoletos puede provocar mas problemas que soluciones. Es necesario considerar también los alcances de la cobertura geográfica de la instalación para que se pueda garantizar el buen flujo de la información.
- **Opciones de expansión.** Contemplar el plan de crecimiento en volumen de información, características adicionales y número de usuarios a corto, mediano y largo plazo.
- **Disponibilidad de aplicaciones y/o herramientas de desarrollo.** La permanencia en el mercado del fabricante y un soporte técnico confiable. El desarrollo de aplicaciones hechas por empresas pequeñas lleva el riesgo de no poder cumplir con las expectativas de servicio adecuado para los requerimientos de la implantación. Adicionalmente, se debe considerar que

la aplicación pueda ser sujeta de modificaciones para los requerimientos propios de la empresa sin depender del proveedor.

- *Costos adicionales.* De instalación y puesta a punto, capacitación a usuarios y al administrador del sistema, mantenimiento, soporte técnico, actualizaciones y expansión. No todos los conceptos que generan gastos son previstos desde el principio, por lo que se debe realizar un análisis antes de hacer la elección. Una buena capacitación a los usuarios y al administrador del sistema minimizan los cargos por soporte técnico y mantenimiento.
- *Compatibilidad con los estándares de Sistemas Abiertos.* Es necesario tener en cuenta los compromisos que se adquieren cuando se implanta un sistema de cómputo con tecnología propietaria sin compatibilidad a otros. Al elegir *hardware* y *software* que cumplen con los estándares de la industria se asegura la libertad de elección en cuanto a fabricantes de opciones de expansión

## *Conclusiones*

No existe una regla general en la elección de un sistema de cómputo para una empresa. Mientras que para unas, las redes de área local son la mejor solución, para otras lo son los sistemas multiusuario.

En ambientes de captura y tareas repetitivas en donde no se requiere manejo de gráficas, los sistemas multiusuario son la mejor elección ya que proporcionan una operación sencilla para el usuario, la capacitación es mínima y el costo de expansión es bajo. Si lo que se requiere son amplias opciones de comunicaciones remotas o la formación de una red de área amplia, los sistemas multiusuario, junto con UNIX, forman un equipo ideal ya que ambos fueron creados para eso.

Por el otro lado, las redes de área local basadas en NetWare son la solución ideal cuando se requiere interconectar las computadoras personales existentes en una empresa para compartir la información y utilizar las aplicaciones ya instaladas. La implantación de aplicaciones gráficas debe considerar costos de *hardware* y *software* para soportar gran tráfico de información en la red. La tendencia actual de trabajar en ambientes de ventanas requiere de mayor dominio de las computadoras por el usuario y es necesaria una capacitación extensa.

A lo largo del trabajo se tocó varias veces el tema de procesamiento distribuido. Esto tiene una razón específica: Las soluciones a través de procesamiento distribuido normalmente toman lo mejor de ambos ambientes, aprovechando las grandes facilidades de manejo de bases de datos demandantes de UNIX y la fácil incorporación de aplicaciones gráficas de NetWare. Ambos ambientes pueden convivir mediante su interconexión en

## **Conclusiones**

red de área local o amplia con protocolos de comunicación estándares como TCP/IP y X.25. Hacer esto no es siempre fácil, ya que debe considerarse la complejidad de la información del usuario, el tiempo de respuesta requerido, el presupuesto asignado y otros factores importantes.

Las líneas de comunicación digital son cada día más accesibles en costo para las empresas por lo que las conexiones remotas entre oficinas centrales y sucursales serán más comunes en corto plazo. Con la tecnología actual es posible incluir datos, imágenes, sonido y video en conexiones remotas, dando mayor posibilidad a las empresas de incrementar su productividad.

Es muy posible que la información de tecnología incluida en este trabajo sea obsoleta cuando llegue a manos del lector, pero lo que siempre será vigente es el hecho de que requeriremos de procesamiento de información mientras exista el ser humano.

## Glosario de Términos

**Análogo/a (Analog)** - Onda o señal continua (por ejemplo, la voz humana).

**Ancho de Banda (Bandwidth)** - Gama de frecuencias que pasan por un circuito. Cuanto mayor el ancho de banda, más información puede enviarse por el circuito en un lapso determinado.

**ANSI (American National Standards Institute)** - Instituto Nacional Estadounidense de Normas.

**ASCII (American Standard Code of Information Interchange)** - Código Estadounidense Normalizado de Intercambio de Información - Código de siete niveles (128 caracteres posibles) con previsión para paridad, usado para la transmisión de datos.

**Baudio** - Unidad de velocidad de señalización equivalente al número de estados o eventos discretos por segundo. Si cada evento de señal representa sólo un estado de bit, la

tasa de baudios equivale a los bps (bits por segundo).

**Backbone.** Tramo principal de una red de área local que tiene gran ancho de banda para soportar el tráfico intenso.

**Bit** - Contracción de *Binary Digit* (Dígito Binario), la menor unidad de información en un sistema binario. Un bit representa o bien uno o cero ("1" o "0").

**Bps (bps - bits per second)** - Bits por segundo. Medida de la velocidad de transmisión de datos en la transmisión serial. Se utiliza también para describir la capacidad de un equipo de transmisión de datos (por ejemplo, un modem de 9600 bps).

**Buffer** - Dispositivo de almacenamiento temporal. Usado normalmente para compensar las diferencias en la velocidad de transmisión de datos o temporización de eventos cuando se transmite de un dispositivo a otro.

**Bus** - Vía o canal de transmisión. Típicamente, un bus es una conexión

## Glosario de términos

eléctrica de uno o más conectores, en el cual todos los dispositivos ligados reciben simultáneamente todo lo que se transmite.

Byte - Grupo de bits que una computadora puede leer (generalmente de longitud de 8 bits).

**C**aracter - Cualquier representación codificada de una letra, número o símbolo especial.

Caracter de Control - Caracteres especiales que permiten el control de dispositivos de comunicación o de impresión.

CCITT (Comité Consultor Internacional de Telegrafía y Telefonía) - Comité asesor internacional, con base en Europa, qurmas internacionales de transmisión.

CD-ROM (*Compact Disk - Read Only Memory*). Dispositivo de almacenamiento masivo por disco compacto de sólo lectura. Provee la capacidad de almacenar gran cantidad de información en un espacio físico pequeño.

Chip. Sinónimo de circuito integrado.

CSMA/CD (*Carrier Sense Multiple Access/Collision Detection*). Método de acceso una red de área local, en el que la contienda entre dos o más estaciones de trabajo se resuelve mediante la detección de choque. Cuando dos estaciones transmiten al mismo tiempo, ambas se detienen y señalan que han tenido un choque. Cada una de ellas trata de transmitir después de un periodo de tiempo predeterminado para evitar el conflicto.

CPU (*Central Processing Unit*), Procesador o Microprocesador. Unidad Central de Proceso, circuito integrado con la capacidad de procesamiento necesaria para dar respuesta a operaciones aritméticas y lógicas, dependiendo de los datos proporcionados.

**D**atos - Información representada en forma digital, incluyendo voz, texto, fax y video.

Digital - La salida binaria ("1/0") de una computadora o terminal. En las

comunicaciones de datos, una señal alternada y discontinua (pulsante).

Dispositivo de Entrada/Salida. Componente de un sistema que permite la entrada o salida de datos para la interacción con el usuario o la comunicación con otros sistemas.

**EBCDIC** (Código Decimal Codificado en Binario Ampliado). Código de caracteres de 8 bits usado principalmente en equipos IBM, adecuado para 256 patrones diferentes.

Estación de trabajo. Computadora personal que se da acceso a un usuario a los recursos de un sistema mayor por medio de líneas de comunicación.

*Ethernet* - Topología o diseño de red de área local normalizado como IEEE 802.3. Utiliza transmisión a 10 Mbps por un bus coaxial, y el método de acceso CSMA/CD. Últimamente adoptado para aplicaciones de estrella sobre par trenzado bajo la norma 10BaseT.

**Host** - Computadora central en una implantación de sistema multiusuario.

**IEEE** (*Institute of Electrical and Electronic Engineers* - Instituto de Ingenieros en Electricidad y Electrónica) - Organización profesional internacional que publica sus propias normas. La IEEE es miembro de ANSI e ISO.

Inodo - En UNIX, es un número de referencia que es asignado a cada archivo del disco para el control interno.

Interface - Límite compartido, definido por características físicas de interconexión en común, características de señal y significados de las señales intercambiadas.

**ISO** (*International Standards Organization* - Organización de Normas Internacional) - Organización internacional involucrada en la formulación de normas de comunicaciones.



**LAN** (*Local Area Network* - Red de Area Local) - Instalación de transmisión de datos de alto volumen que conecta varios dispositivos intercomunicados (computadoras, terminales e impresoras) dentro de una misma habitación, edificio o complejo.

**Linker.** Módulo de un sistema operativo que establece los enlaces de los programas de aplicación con las utilerías del sistema operativo.

**Mainframe.** Sistemas multiusuario y centralizado de gran poder de cómputo para grandes corporativos.

**Memoria física.** Memoria que se encuentra integrada en el *hardware* de una computadora con el propósito de almacenar datos. Su limitante de almacenamiento lo establece la cantidad y tipo de circuitos integrados de memoria que se instalan.

**Memoria virtual.** Combinación de la memoria física y el uso de discos duros para almacenamiento temporal de datos.

**Minicomputadora.** Sistema multiusuario utilizado para aplicaciones demandantes de empresas medianas.

**Modem (Modulador - Demodulador)** - Dispositivo utilizado para convertir señales digitales en serie de un dispositivo transmisor a una señal adecuada para la transmisión por línea telefónica. Reconvierte también la señal transmitida a información digital serie para su aceptación por un dispositivo receptor.

**MS-DOS** (*Microsoft Disk Operating System*). Sistema operativo monousuario y monotarea desarrollado por Microsoft para computadoras personales con procesador Intel.

**Nodo** - Punto de interconexión a una red.

**OSI Model** (*Open Systems Interconnection*) - Modelo de referencia de siete capas de red de comunicaciones desarrollado por la ISO.

**Paquete (Packet)** - Grupo ordenado de señales de datos y de control transmitido por una red y que es un subconjunto de un mensaje más grande.

**Par Trenzado con Blindaje (STP, Shielded Twisted Pair)** - Término general que designa sistemas de cableado específicamente diseñados para la transmisión de datos y en los cuales los cables llevan una protección (blindaje) contra ruido eléctrico.

**Par Trenzado (Twisted Pair)** - Un par de hilos trenzados para la transmisión de datos.

**Par Trenzado sin Blindaje (UTP, Unshielded Twisted Pair)** - Término general que designa sistemas locales de cableado específicamente diseñados para la transmisión de datos y que no están blindados.

**Polling (Escrutinio).** Técnica de comunicaciones que determina cuándo una terminal está lista para enviar datos. La computadora central interroga continuamente a todas las terminales conectadas a ella en una secuencia de circuito cíclico. Una

terminal reconoce el escrutinio que tiene dato para ser enviados.

**Protocolo** - Conjunto formal de convenciones que gobiernan el formato y temporización relativa del intercambio de mensajes entre dos sistemas que se comunican.

**Puente (Bridge)** - Dispositivo que interconecta redes de áreas local (LAN) en la capa de enlace de datos OSI.

**Puerto (Port)** - Interface físico a una computadora para la conexión de dispositivos terminales o de comunicación.

**Red (Network)** - (1) Grupo de nodos interconectados (2) Serie de puntos, nodos o estaciones conectados por canales de comunicación; (3) El conjunto de equipos por medio del cual se establecen las conexiones entre las estaciones de datos.

**Time-Sharing (Tiempo-Compartido).** Método de operación computacional que permite a varias terminales

interactivas unar una computadora central. Aún cuando las terminales son realmente servidas en secuencia, la alta velocidad de la computadora hace aparecer esto como si todas las terminales fueran servidas simultáneamente.

*Token Ring* - Mecanismo de acceso a una red de área local y topología en la cual una trama supervisora (*token*) es pasada secuencialmente entre estaciones adyacentes. Las estaciones que desean acceder a la red deben esperar la llegada del *token* antes de poder transmitir datos. En un *Token Ring*, la próxima estación lógica que recibe el *token* es también la próxima estación física en el anillo.

Tráfico de red. Saturación de la línea de comunicación por la cantidad de información transmitida.

Transmisión asíncrona - Método de transmisión el cual envía las unidades de datos de a un caracter a la vez. Los caracteres son precedidos y seguidos por bits de arranque/parada (*start/stop*) que dan la temporización (sincronización) en la terminal receptora. Llamada también transmisión de arranque/parada.

Transmisión serial - El modo de transmisión más común, en el cual los bits de los caracteres son enviados secuencialmente a uno por vez en lugar de en forma paralela.

Transmisión sincrona - Transmisión en la cual los bits de datos se envían a velocidad fija, con el transmisor y receptor sincronizados. La transmisión sincrona hace innecesarios los bit de arranque/parada.

**U**suario. Cualquier personal que interactúa con una computadora a través de cualquier dispositivo de Entrada/Salida.

## Bibliografía

Andleigh, Prabhat K.  
UNIX systems architecture  
Prentice-Hall, Inc.  
1ª Edición, U.S.A., 1990

Stallings, William  
Operating systems  
Macmillan Publishing Company  
1ª Edición, U.S.A., 1992

Michael, Day  
Enterprise Series: Downsizing to NetWare  
New Riders Publishing  
1ª Edición, U.S.A., 1992

Padovano, Michael  
Networking applications on UNIX System V release 4  
Prentice-Hall, Inc.  
1ª Edición, U.S.A., 1993

Heywood, Drew/Jerney, Janos (John)/Johnston, Jon  
Enterprise Series. Connectivity: Local Area Networks  
New Riders Publishing  
1ª Edición, U.S.A., 1992

SCO UNIX System V/386 Operating System,  
System Administrator's Guide  
The Santa Cruz Operation, Inc.  
3ª Edición, U.S.A., 1990

Gofon, Peter W.  
Mastering UNIX Serial Communications  
Sybex, Inc.  
1ª Edición, U.S.A., 1991

The Basics Book of Frame Relay  
Motorola University Press  
Corporate & Professional Publishing Group  
Addison-Wesley Publishing Company  
2ª Edición, U.S.A., 1992

Edgar B. Villagómez Valdés

*Bibliografía*

**The Basics Book of Information Networking**  
Motorola University Press  
Corporate & Professional Publishing Group  
Addison-Wesley Publishing Company  
1ª Edición, U.S.A., 1991

**The Basics Book of X.25 Packet Switching**  
Motorola University Press  
Corporate & Professional Publishing Group  
Addison-Wesley Publishing Company  
7ª Edición, U.S.A., 1996

**SCO System V Network Administration, Student Guide**  
The Santa Cruz Operation, Inc.  
4ª Edición, U.S.A., 1991

**SCO System V/386 Administration, Student Guide Vol. 1 y 2**  
The Santa Cruz Operation, Inc.  
3ª Edición, U.S.A., 1991