


MÓDULO 2 MF0964_3: DESARROLLO DE ELEMENTOS DE SOFTWARE PARA GESTIÓN DE SISTEMAS

1 Descripción de los servicios,
estructura y administración de
sistemas operativos

CORE
networks




1 Descripción de los servicios, estructura y administración de sistemas operativos

1.1 Introducción

CORE
networks

Sistema operativo

Un sistema operativo es el software principal o conjunto de programas de un sistema informático que gestiona los recursos de hardware y provee servicios a los programas de aplicación de software, ejecutándose en modo privilegiado respecto de los restantes.



1 Descripción de los servicios, estructura y administración de sistemas operativos

1.2 Definición y conceptos básicos sobre sistemas operativos

CORE
networks

Bloques funcionales

EJECUCIÓN DE PROGRAMAS Y GESTIÓN DE PROCESOS

- Procesos de arranque y finalización del sistema
- Gestión de memoria
- Gestión de red
- Gestión de almacenamiento (sistema de archivos)
- Gestión de dispositivos de Entrada/Salida
- Gestión de errores
- Auditoría (logs del sistema)

Tareas de cada bloque funcional

- Secuenciación
- Concurrencia
- Permisos
- Recursos
- Interrupciones
- Errores

Gestión de la memoria

La gestión de memoria o administración de memoria es el acto de gestionar la memoria de un dispositivo informático, asignando la misma a los programas que la solicitan. Los sistemas de memoria virtual separan las direcciones de memoria utilizadas por un proceso de las direcciones físicas reales, permitiendo la separación de procesos e incrementando la cantidad efectiva de memoria de acceso aleatorio utilizando la paginación.

El recolector de basura es la asignación y liberación automática de los recursos de memoria para un programa. La implementación suele ser a nivel del lenguaje de programación en contraposición a la gestión manual de memoria, que asigna y libera los recursos de memoria de una computadora de forma explícita.

Memoria virtual

Gracias a la memoria virtual, un sistema operativo moderno presenta al usuario un espacio de memoria contiguo que puede ser mayor que la cantidad de memoria principal instalada.

Esto se consigue al utilizar dinámicamente memoria secundaria para guardar información que aparentemente está en memoria principal (cuando se necesita dicha información hay que llevarla de nuevo a memoria principal).

Al espacio de memoria secundaria que guarda datos que aparentan estar en memoria principal se le llama espacio de trasiego o intercambio, en el caso de GNU/Linux, espacio swap.

Memoria virtual (II)

El uso de memoria virtual:

- Permite que la cantidad de espacio dedicado a los procesos no esté limitado por la cantidad de memoria principal instalada. Es posible tener cargados más procesos y es posible cargar procesos más grandes.
- Reduce la velocidad de ejecución. Cuando se ejecutan procesos que corren en memoria secundaria no se puede igualar la velocidad de proceso que se conseguiría si se utilizase memoria principal.
- El SO debe gestionar la traducción de direcciones entre el espacio de memoria virtual y físico. Para este fin el SO se apoya en hardware específico como la MMU (unidad de gestión de memoria, normalmente integrada en la CPU).

Memoria virtual (III)

Recurso

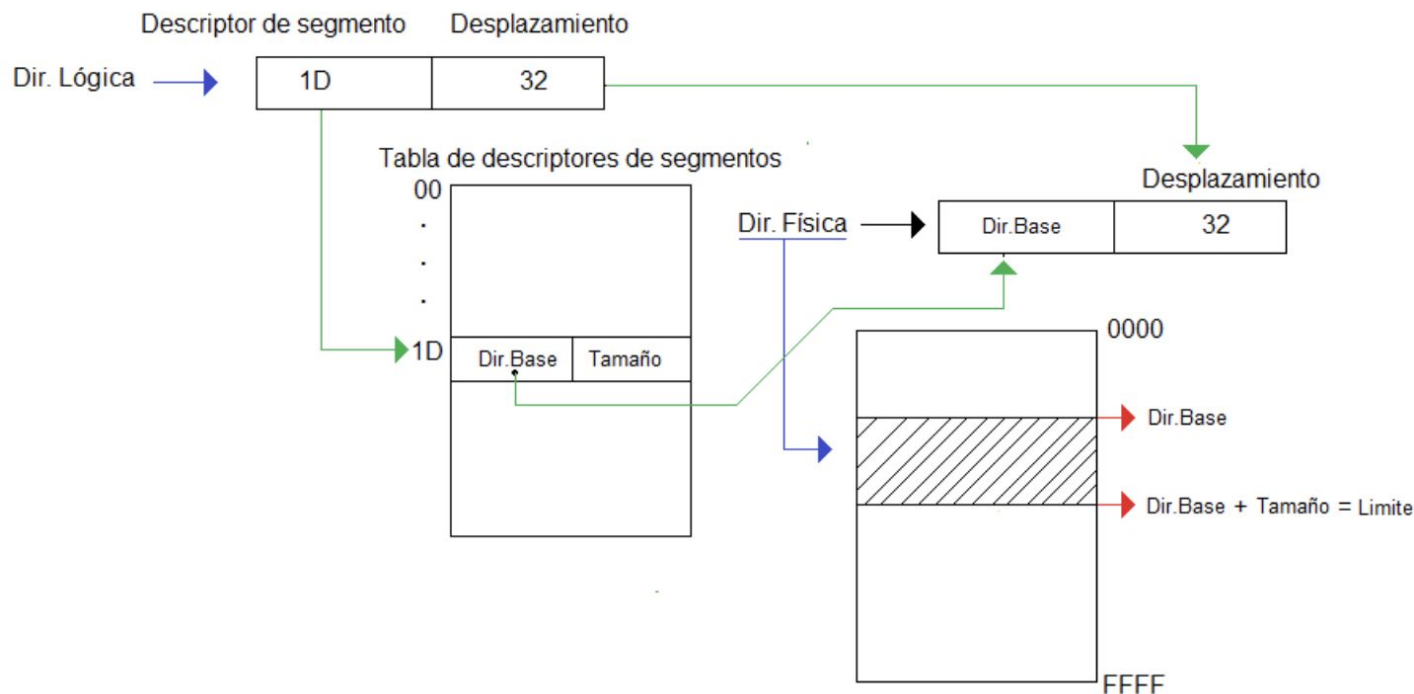
<https://support.microsoft.com/es-es/help/2160852/ram-virtual-memory-pagefile-and-memory-management-in-windows>

Segmentación

Un segmento es un espacio de memoria de tamaño variable, compuesto por:

- Descriptor: Identificador único del segmento (dentro del espacio de memoria del proceso).
- Tamaño del segmento

Segmentación (II)



Segmentación (III)

Características:

- Cada proceso en ejecución (esté activo, bloqueado o preparado) tiene su tabla de segmentos.
- Solapamiento: Se puede hacer que 2 segmentos se superpongan de manera que compartan direcciones de memoria física con direcciones lógicas diferentes. Para evitar problemas de concurrencia, debe indicarse explícitamente que una porción de memoria pueda ser compartida. De esta manera, procesos diferentes pueden compartir información y código usando la memoria común.
- Cuando un proceso requiere más memoria se crea un nuevo segmento.

Segmentación (IV)

- Dos instancias de un mismo proceso pueden compartir segmentos de memoria de instrucciones/código, pero no para datos ya que esto complicaría la gestión.
- Es posible la redimensión de segmentos siempre que haya posiciones libres contiguas, o crear un nuevo segmento y copiar el contenido del anterior.
- Gestión compleja, sobre todo por su tamaño variable.

Segmentación (V)

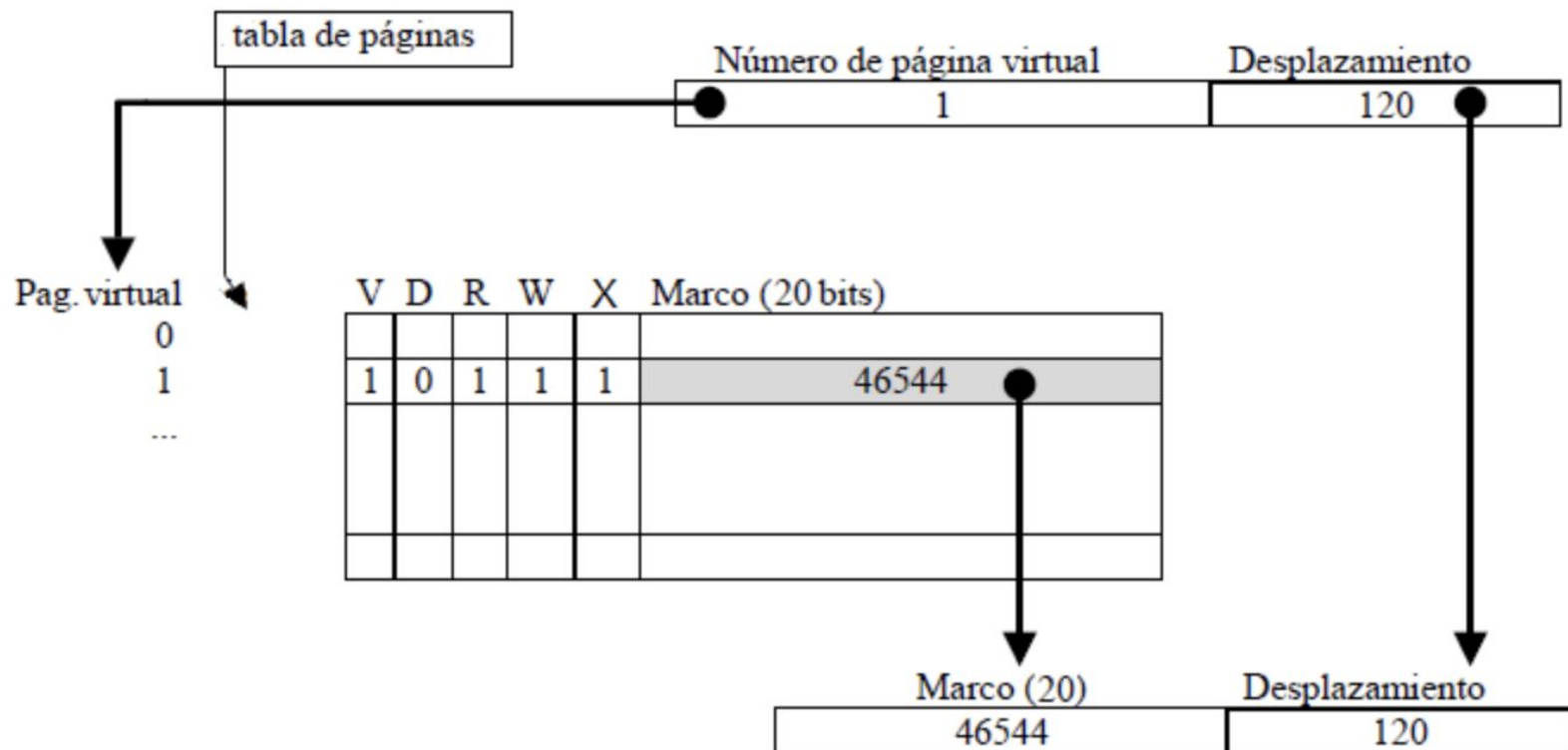
- Permite la carga de segmentos a petición, de manera que no se disponga de todos los segmentos en memoria principal, que se puedan descargar a disco (en la zona de intercambio o swap) en base a un cierto criterio (ver Memoria virtual). Esta zona de intercambio puede ser:
 - Un lugar fijo en el disco (Linux, Unix). El administrador de memoria decide qué segmento se va a descargar
 - Un fichero oculto de tamaño fijo o variable dependiendo de la configuración dada por el usuario (Windows).
- La segmentación se hizo para equipos con poca memoria, no está pensada para sistemas modernos.

Paginación

La paginación es una estrategia de organización de la memoria física que consiste en dividir la memoria en porciones de igual tamaño. A dichas porciones se las conoce como páginas físicas o marcos. Los marcos están identificados por un número, conocido como número de página física. Cada página física se asigna en exclusividad a un proceso.

A su vez, se dispone de páginas lógicas. Todo proceso tiene su propio espacio de páginas lógicas, cada página lógica está mapeada a un marco y este mapeo entre páginas lógicas y páginas físicas se mantiene en la tabla de páginas. Cada proceso dispone de su propia tabla de páginas. En tiempo de conmutación de procesos, se restablece la tabla de páginas del proceso que pasa a estado a activo.

Paginación (II)



Paginación (III)

Las páginas disponen de mecanismos de protección, expresados en bits de lectura, escritura y ejecución. Lo que permite definir páginas de sólo lectura, por ejemplo, para almacenar constantes. Páginas de sólo lectura y ejecución, para almacenar código, así como páginas de lectura y escritura para el heap (montón) o la pila (stack) del proceso.

¿Cuál es el tamaño de página idóneo? Los valores óptimos para las porciones son: en arquitectura de 32 bits es de 4KB, y en arquitectura de 64 bits es de 8KB.

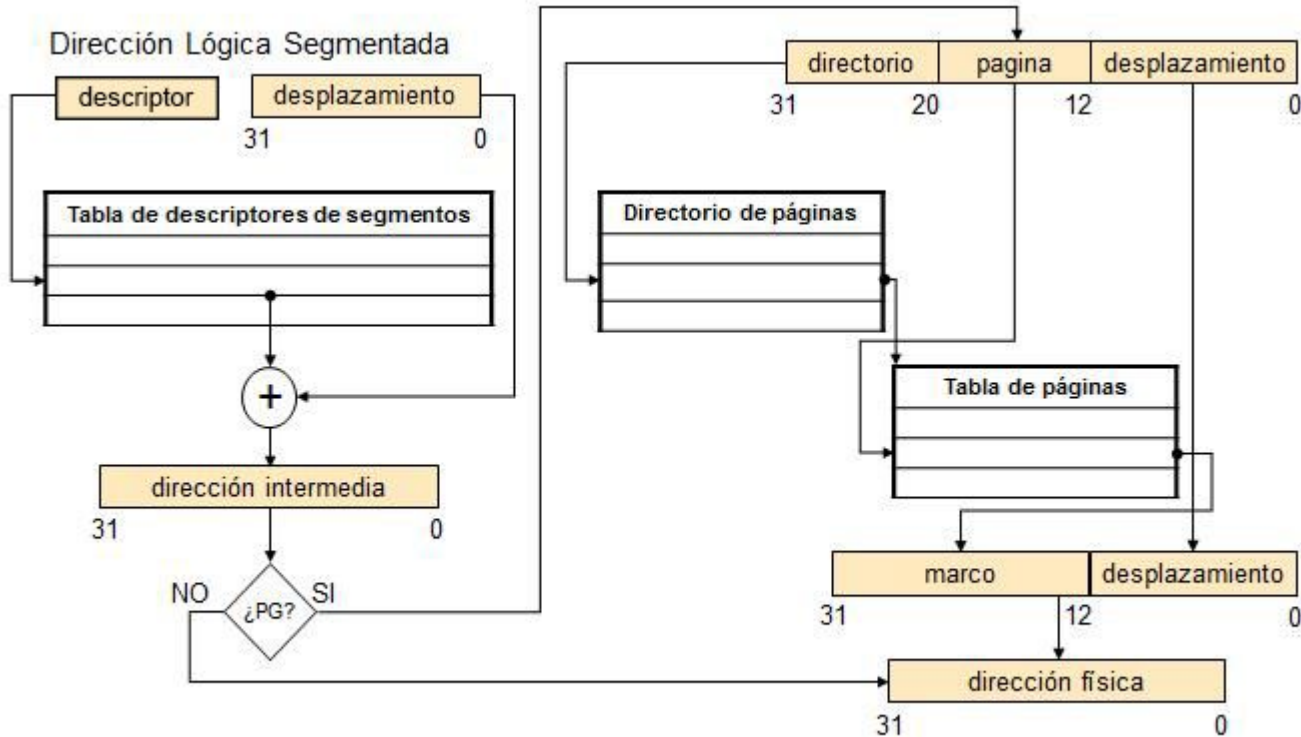
Los criterios vistos en la gestión de memoria mediante particiones de tamaño variable no tienen aplicación en la gestión de la memoria paginada, pues todos los huecos disponibles son de igual tamaño.

Segmentación paginada

Consiste en combinar la segmentación y la paginación. La idea fundamental consiste en definir segmentos que están compuestos por N páginas. De ahí que reciba el nombre de segmentación paginada. Por tanto, cualquier segmento ocupa, como mínimo una página de memoria.

Por una parte, la segmentación proporciona ventajas hacia los procesos: organización de datos y código, mecanismos de protección, compartición de datos y código, y optimización en el espacio ocupado. Por otra parte, la paginación facilita la gestión de la memoria disponible.

Segmentación paginada (II)



Segmentación paginada (III)

Los procesos emplean direcciones lógicas de memoria segmentada, que son traducidas a direcciones lógicas paginadas (dirección intermedia), que a su vez deben de ser traducidas a direcciones físicas. Por tanto, cualquier acceso a memoria supone un total de tres accesos a memoria. Uno para acceder a la tabla de segmentos, otro para acceder a la tabla de páginas y, por último, otro más para acceder a la dirección física.

Segmentación paginada (III)

Los procesos emplean direcciones lógicas de memoria segmentada, que son traducidas a direcciones lógicas paginadas (dirección intermedia), que a su vez deben de ser traducidas a direcciones físicas. Por tanto, cualquier acceso a memoria supone un total de tres accesos a memoria. Uno para acceder a la tabla de segmentos, otro para acceder a la tabla de páginas y, por último, otro más para acceder a la dirección física.

Memoria virtual

La memoria virtual para aumentar el grado de multiprogramación. Para ello se emplea una zona en disco conocida como zona de intercambio o swap para descargar en ella segmentos o páginas que se encuentran en memoria principal.

Memoria virtual (II)

La zona de intercambio se implementa siguiendo dos posibles estrategias:

- Se destina una parte del disco reservada explícitamente para ello. En este caso, dicho espacio de memoria no puede ser empleado para almacenar información útil del usuario. Los sistemas operativos tipo-Unix implementan esta estrategia.
- Se emplea un fichero oculto que puede tener un tamaño fijo (donde dicho espacio estará reservado y el usuario no podrá hacer uso de él) o variable para almacenar las páginas descargadas. Los sistemas operativos de tipo Microsoft implementan esta aproximación.

Memoria virtual (III)

La técnica de memoria virtual se manifiesta cuando toda la memoria principal está en uso y, por tanto, no existe espacio libre para cargar una cierta página que es requerida por un proceso.

Ante esta situación, el mecanismo de memoria virtual tiene que recargar la página que se necesita en la memoria principal, de lo contrario debe seleccionar una página víctima que será descargada a la zona de intercambio para dejar espacio a la que se necesita. La selección de las páginas víctima se hace en base a diferentes criterios que veremos en esta unidad didáctica.

Memoria virtual (IV)


Por tanto, en caso de que un proceso activo requiera una página o segmento que está descargado en la zona de intercambio, hay que seleccionar una página víctima de ese mismo proceso (si se emplea un criterio de selección local) o de otro proceso (si se emplea uno global), preferentemente de un proceso en estado preparado o bloqueado. La página víctima se descarga en disco para hacer espacio a la página requerida.

Los tiempos de descarga y recarga de páginas tienen una penalización considerable, pues es mayor el tiempo de acceso que necesita un disco magnético que la memoria principal. En general, un buen criterio de reemplazo de páginas debe no descargar páginas que vayan a ser necesitadas en poco tiempo, reduciendo así la cantidad de accesos a disco asociados a la descarga y recarga de páginas.

Logs

En informática, se usa el término log, historial de log o registro, se refiere a la grabación secuencial en un archivo o en una base de datos de todos los acontecimientos que afectan a un proceso particular.

De esta forma constituye una evidencia del comportamiento del sistema. En los sistemas modernos los logs se almacenan persistentemente lo que permite emplearlos a su vez como funcionalidad de negocio además de su valor intrínseco en las tareas de administración.



1 Descripción de los servicios, estructura y administración de sistemas operativos

1.3 Características estructurales de los sistemas operativos

CORE
networks

Modos de operación del procesador

El modo de ejecución del procesador determina qué conjunto de instrucciones y a qué recursos del procesador se puede acceder.

Los procesadores ofrecen, como mínimo, dos modos de funcionamiento del procesador:

- Modo privilegiado, que permite la ejecución de todo el conjunto de instrucciones que ofrece el procesador. Es el modo en el que se ejecuta el núcleo de un sistema operativo.
- Modo no privilegiado, que permite acceder a un subconjunto de las instrucciones que ofrece el procesador, es el modo en el que se ejecutan los procesos.

Núcleo del sistema operativo

El núcleo del sistema operativo, también conocido por la terminología inglesa *kernel*, es la parte más esencial del sistema operativo. Es la parte del código del sistema operativo que se ejecuta en modo privilegiado del procesador.

Al operar en dicho modo, un error de programación en el núcleo del sistema operativo resulta en un error fatal del cual el sistema sólo puede recuperarse mediante el reinicio del sistema.

A tal error fatal también se le conoce en los sistemas operativos UNIX por la locución inglesa *kernel panic* y en los sistemas operativos Windows por el nombre de Pantallazo azul o BSOD.

Tipos de Sistemas Operativos

Los sistemas operativos se pueden clasificar en base a la cantidad de funcionalidades propias del sistema operativo que se implementan en su núcleo.

En general distinguimos dos tipos de sistemas operativos:

- Monolíticos: gestionan los cuatro componentes fundamentales del sistema operativo en modo supervisor, esto incluye, la planificación de procesos, la administración de la memoria principal, la gestión de E/S y el administrador de ficheros.

Tipos de Sistemas Operativos (II)

- Micronúcleos: implementan en su núcleo únicamente la planificación de procesos, la gestión de interrupciones (la parte básica fundamental de la gestión de E/S que necesariamente se tiene que realizar en modo privilegiado) y la comunicación entre procesos. Por tanto, la administración de memoria principal, la gestión de la E/S y la gestión de ficheros se realiza en modo usuario. En este tipo de sistema operativo hay procesos especiales propios del sistema operativo que implementan dichas funcionalidades en modo usuario y se denominan proceso servidor.

Tipos de Sistemas Operativos (III)

Además, contamos con una clasificación intermedia:

- Híbrido: a mitad de camino entre monolítico y micronúcleo, incluyen la gestión de dispositivos de E/S en modo supervisor con el fin de mejorar el rendimiento del sistema.

Los sistemas operativo monolíticos ofrecen mejor rendimiento que los micronúcleos, pero los micronúcleos son más fiables, pues un fallo de programación en el núcleo lleva al traste con la ejecución del sistema.

Sistemas operativos Monolíticos

Los sistemas operativos monolíticos se caracterizan por implementar en el núcleo los cuatro componentes fundamentales del sistema operativo, que son la planificación de procesos, la administración de la memoria principal, la administración de ficheros y la gestión de los dispositivos de entrada/salida.

Los sistemas operativos de propósito general son predominantemente monolíticos hoy día, algunos ejemplos son:

- Sistemas operativos UNIX, tales como FreeBSD, NetBSD y OpenBSD.
- Sistemas operativos GNU/Linux, y por tanto, Android también.
- DOS, tales como MS-DOS y DR-DOS.

Sistemas operativos Monolíticos (II)

Como inconveniente, este tipo de sistemas operativo dispone de un alto número de líneas de código ejecutándose en modo privilegiado. Por ello, un error de programación en el núcleo puede provocar un kernel panic. Además, el hecho de añadir nuevas funcionalidades provocaría una nueva recompilación del núcleo llevando a reiniciar el sistema para que se apliquen los nuevos cambios. Por ejemplo, un bug en el núcleo de un sistema operativo monolítico obliga a tener que reiniciar el sistema.

Como principal ventaja, los sistemas operativos monolíticos ofrecen un alto rendimiento puesto que las peticiones entre los diferentes componentes se reducen a invocaciones de funciones.

Sistemas operativos Micronúcleo

La principal ventaja de los sistemas operativos micronúcleo es que, al ejecutar menos líneas de código en modo privilegiado, de manera intuitiva son más fiables. Otras ventajas son que se garantiza el aislamiento de las partes que están fuera del núcleo. Generalmente, si un proceso servidor tiene un bug en su código que hace que entre en una condición de error, se puede relanzar sin tener que reiniciar el sistema por completo.


Sin embargo, el principal problema que presentan es el rendimiento, puesto que cualquier petición requiere mensajería, que lleva consigo un coste extra debido a la construcción de los mensajes, el reparto y la interpretación. Son estos problemas relacionados con el rendimiento los que hacen que no existan sistemas operativos micronúcleo, y que generalmente únicamente tiene propósitos educativos y/o propósitos muy específicos.

Sistemas operativos Híbridos

Híbrido implica que el núcleo en cuestión usa conceptos de arquitectura tanto del diseño monolítico como del micronúcleo, específicamente el paso de mensajes y la ejecución de ciertos componentes del sistema operativo en espacio de usuario.

Algunos ejemplos de núcleos híbridos:

- Microsoft Windows NT, usado en todos los sistemas que usan el código base de Windows NT.
- XNU (usado en Mac OS X), es un micronúcleo modificado, debido a la inclusión de código del núcleo de FreeBSD en el núcleo basado en Mach.
- DragonFlyBSD, es el primer sistema BSD que adopta una arquitectura de núcleo híbrido sin basarse en Mach.
- ReactOS.



1 Descripción de los servicios, estructura y administración de sistemas operativos

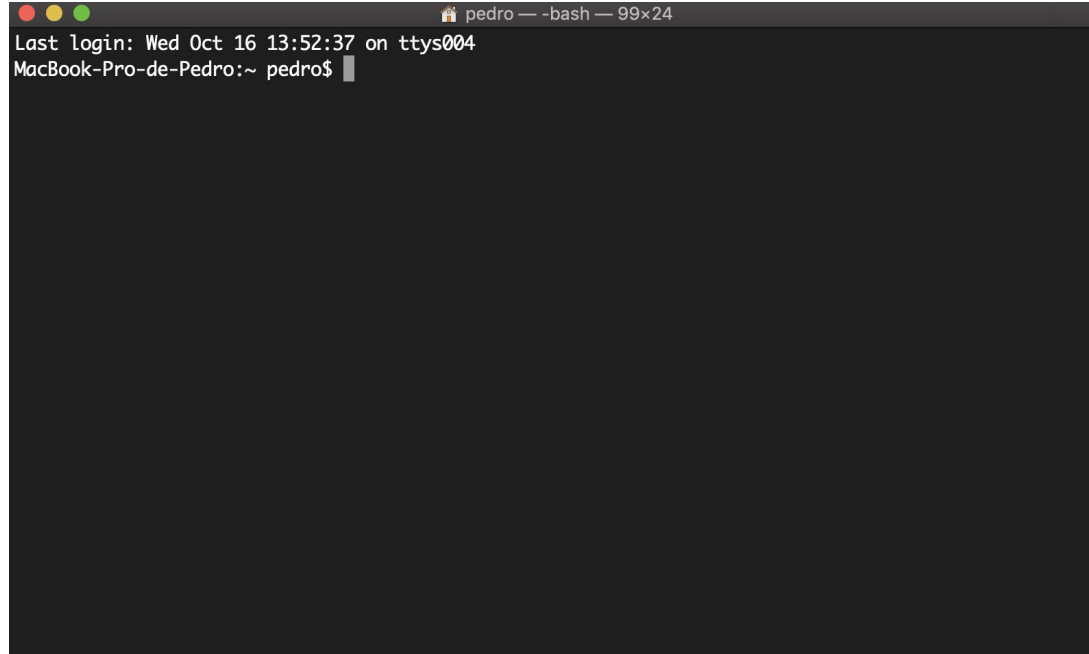
1.4 Herramientas administrativas de uso común en sistemas operativos

CORE
networks

Interfaces gráficas de usuario (GUI)




Herramientas de línea de comando (CLI)

A screenshot of a macOS terminal window. The title bar at the top shows a home icon, the name 'pedro', and the command '-bash' followed by the window size '99x24'. The terminal content shows a login message: 'Last login: Wed Oct 16 13:52:37 on ttys004'. Below this, the prompt 'MacBook-Pro-de-Pedro:~ pedro\$' is displayed with a cursor at the end.

```
pedro — -bash — 99x24
Last login: Wed Oct 16 13:52:37 on ttys004
MacBook-Pro-de-Pedro:~ pedro$
```

Nuevas interfaces





1 Descripción de los servicios, estructura y administración de sistemas operativos

1.5 Resumen

CORE
networks

Resumen

- Definición y conceptos básicos sobre sistemas operativos.
- Características estructurales de los sistemas operativos.
- Herramientas administrativas de uso común en sistemas operativos.