INFORMÁTICA

Trabajo Práctico N°1 – Introducción a la arquitectura de computadoras y sistemas operativos.

Videla Jeronimo 14233

Parte 2: Introducción a los Sistemas Operativos

1) **¿Qué es un sistema operativo y cuáles son sus principales funciones?**
   Un sistema operativo es un software que actúa al nivel del hardware y tiene dos funciones principales: Proveer de una maquina virtual al usuario y ser un controlador de los recursos de cómputo.

2) **Explique qué es una llamada al sistema y proporcione un ejemplo.**
   Para obtener servicios del sistema operativo, un programa usuario debe lanzar una llamada al sistema, la cual se atrapa en el kernel e invoca al sistema operativo. La interfaz entre el SO y los programas de usuario es el conjunto de llamadas al sistema.
   Ejemplo de una llamada al sistema: write()
   Supongamos que un programa en C quiere escribir una cadena de texto en la consola. Puede usar la llamada al sistema write() para hacerlo.

3) **Verdadero/Falso (Justificar las falsas):**
   a. Un proceso siempre está asociado a un único hilo de ejecución.
      Falso.
      En los sistemas operativos modernos, un proceso puede estar asociado a múltiples hilos de ejecución. Esto es lo que se conoce como multithreading, donde un solo proceso puede ejecutar múltiples hilos de forma concurrente. Cada hilo dentro del proceso comparte el mismo espacio de memoria y recursos, pero pueden ejecutarse de manera independiente.

   b. Los sistemas operativos modernos permiten la ejecución de múltiples procesos simultáneamente.
      Verdadero.

4) **¿Qué es la multiprogramación?**
   La multiprogramación es una técnica utilizada en los sistemas operativos para mejorar la utilización de la CPU al permitir que varios programas o procesos residan en la memoria principal (RAM) al mismo tiempo. La idea principal es maximizar el uso del tiempo de CPU al mantenerlo ocupado con trabajo útil en lugar de estar inactivo.
   El CPU solamente puede ejecutar una tarea a la vez, pero los tiempos de ejecución son tan rápidos que se crea la ilusión de paralelismo.

5) **Describa el concepto de memoria virtual y su importancia en los sistemas operativos modernos.**
   La memoria virtual proporciona la habilidad de ejecutar programas más extensos que la memoria física de la computadora, llevando y trayendo pedazos entre la RAM y el

disco. La memoria virtual también permitió la capacidad de ligar dinámicamente un programa a una biblioteca en tiempo de ejecución, en vez de compilarlo.

6) **Dibuje un diagrama que muestre los estados de un proceso y las transiciones entre estos estados.**

1. El proceso se bloquea para recibir entrada
2. El planificador selecciona otro proceso
3. El planificador selecciona este proceso
4. La entrada ya está disponible

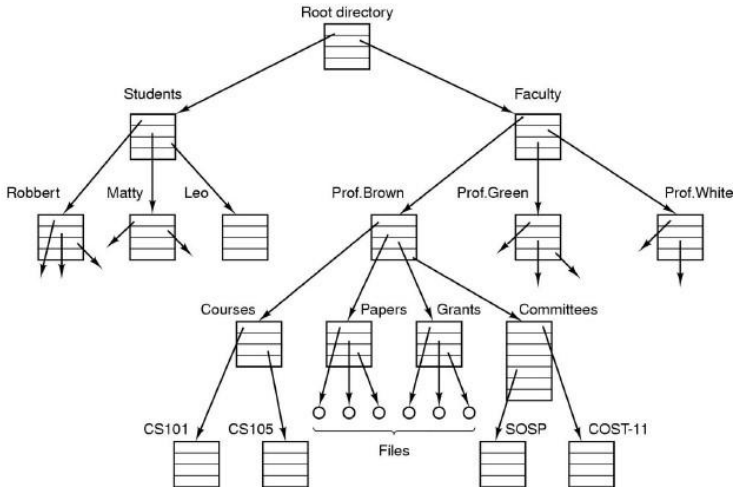
Figura 2-2. Un proceso puede encontrarse en estado "en ejecución", "bloqueado" o "listo". Las transiciones entre estos estados son como se muestran.

7) **¿Para qué sirve la estructura de directorios?**

El directorio tiene por finalidad ubicar el contenido de los archivos que se encuentran almacenados en el disco

En la mayoría de los sistemas, cada entrada de directorio contiene una lista de los archivos allí almacenados, y un apuntador a la posición de estos en el disco

8) **De la ruta de acceso de:**



a. El directorio CS101 (ruta de acceso absoluta).
Root/Faculty/Prof.Brown/Courses/CS101

b. Un archivo llamado Paper1, dentro del directorio Papers. El directorio actual es Faculty (ruta de acceso relativa).
Prof.Brown/Papers/Paper1

9) **Explique los permisos de archivos en un sistema Unix (rwx) ¿De que manera se pueden configurar? De un ejemplo de cada una.**

En un sistema Unix o Linux, los permisos de archivos son fundamentales para controlar el acceso a los archivos y directorios. Estos permisos se representan con una

combinación de letras y números y especifican qué acciones pueden realizarse sobre un archivo o directorio.

Tipos de Permisos

r (read) - Lectura:

- Permite leer el contenido del archivo o, en el caso de un directorio, listar su contenido.
- Ejemplo: r-- significa que solo se tiene permiso de lectura.

w (write) - Escritura:

- Permite modificar el contenido del archivo o, en el caso de un directorio, añadir, eliminar o renombrar archivos dentro de él.
- Ejemplo: -w- significa que solo se tiene permiso de escritura.

x (execute) - Ejecución:

- Permite ejecutar el archivo si es un programa o script. En un directorio, permite acceder a los archivos y subdirectorios.
- Ejemplo: --x significa que solo se tiene permiso de ejecución.

`chmod u+rwx file`: Otorga al propietario permisos completos (lectura, escritura, ejecución) sobre el archivo.

`chmod 755 script.sh`: Da permisos de ejecución al propietario, y permisos de lectura y ejecución al grupo y a otros.

`chmod g-w file`: Revoca el permiso de escritura para el grupo en el archivo.

10) ¿Qué es una tubería (pipe) y para qué se utiliza en los sistemas operativos?
Un canal (pipe) es un tipo de pseudoarchivo que puede utilizarse para conectar dos procesos. Si los procesos A y B desean comunicarse mediante el uso de un canal, deben establecerlo por adelantado. Cuando el proceso A desea enviar datos al proceso B, escribe en el canal como si fuera un archivo de salida. El proceso B puede leer los datos a través del canal, como si fuera un archivo de entrada. Por ende, la comunicación entre procesos en UNIX tiene una apariencia muy similar a las operaciones comunes de lectura y escritura en los archivos. Y por si fuera poco, la única manera en que un proceso puede descubrir que el archivo de salida en el que está escribiendo no es en realidad un archivo sino un canal, es mediante una llamada al sistema especial.

11) **Verdadero/Falso (Justificar las falsas):**
    a. La segmentación y la paginación son técnicas de administración de la memoria utilizadas para evitar la fragmentación interna.
    Falso.
    Paginación es una técnica de administración de memoria que divide la memoria física en bloques de tamaño fijo llamados páginas. La paginación está diseñada para evitar la fragmentación externa, no la interna. La fragmentación interna puede ocurrir en la paginación cuando una página no se llena completamente.
    Segmentación divide la memoria en segmentos de diferentes tamaños que corresponden a diferentes tipos de datos o programas. La segmentación también ayuda a evitar la fragmentación externa al permitir que cada segmento crezca o se reduzca independientemente. Sin embargo, la

segmentación, por sí sola, no aborda la fragmentación interna de manera directa.

b. Un directorio puede contener otros directorios y archivos en un sistema de archivos jerárquico.
Verdadero

12) **Investigue y describa las características de un sistema operativo de tiempo real (RTOS) utilizado en la industria.**
Un Sistema Operativo de Tiempo Real (RTOS) es un tipo de sistema operativo diseñado para ejecutar tareas dentro de plazos específicos y predecibles.
Sus características clave son:
1. Determinismo: Respuesta predecible y mínima latencia.
2. Planificación de Tareas: Gestión de tareas mediante prioridades, permitiendo interrumpir tareas menos urgentes.
3. Manejo de Interrupciones: Procesa eficientemente las interrupciones de hardware.
4. Gestión de Memoria: Prefiere memoria estática para asegurar acceso rápido y evitar fragmentación.
5. Multitarea: Permite la ejecución simultánea de múltiples tareas con prioridad.
6. Gestión de Recursos: Sincroniza y controla el acceso a recursos compartidos, evitando problemas de concurrencia.
7. Servicios de Comunicación: Proporciona mecanismos como colas de mensajes para la comunicación entre tareas.
8. Sistemas de Archivos: A veces incluye sistemas de archivos optimizados para tiempo real.
9. Tamaño y Overhead: El núcleo es pequeño y optimizado para sistemas con recursos limitados.

Ejemplos de RTOS en la Industria:
- FreeRTOS: Un RTOS de código abierto ampliamente utilizado en sistemas embebidos.
- VxWorks: Utilizado en aplicaciones críticas y de alta disponibilidad, como en el espacio y la defensa.
- QNX: Ofrece alta disponibilidad y es utilizado en automóviles y sistemas industriales.
- RTEMS (Real-Time Executive for Multiprocessor Systems): Usado en aplicaciones embebidas, especialmente en sistemas de vuelo y espaciales.

13) **¿Cómo se gestionan los dispositivos de E/S en un sistema operativo? Explique las capas involucradas en este proceso.**
La gestión de dispositivos de entrada/salida (E/S) en un sistema operativo se organiza en varias capas clave:
1. Controladores de Dispositivos: Programas que permiten al sistema operativo comunicarse directamente con el hardware, proporcionando una interfaz uniforme.
2. Gestión de E/S: Coordina las operaciones de E/S, incluyendo el buffering, caching, y la planificación de solicitudes para optimizar el rendimiento.

3. Sistemas de Archivos: Proporciona una interfaz para acceder a archivos, abstrayendo los detalles del almacenamiento físico.
4. Interfaz de Usuario: Permite al usuario interactuar con los dispositivos de E/S mediante comandos o interfaces gráficas.

Este diseño en capas facilita la interacción eficiente entre software y hardware, permitiendo flexibilidad y extensibilidad en la gestión de dispositivos.

14) **Describa los principales sistemas de archivos utilizados en los sistemas operativos modernos y sus diferencias.**

NTFS (Windows)

Características: Seguridad avanzada, journaling, soporta archivos grandes y volúmenes grandes.

Uso: Principal en Windows.

FAT32

Características: Amplia compatibilidad, limita archivos a 4 GB y volúmenes a 8 TB.

Uso: Dispositivos portátiles y sistemas antiguos.

exFAT+

Características: Soporta archivos grandes (más de 4 GB), compatible con Windows y macOS.

Uso: Dispositivos portátiles modernos.

EXT4 (Linux)

Características: Buen rendimiento, journaling, soporta archivos grandes y volúmenes grandes.

Uso: Principal en Linux.

APFS (macOS)

Características: Optimizado para SSD, soporta snapshots, cifrado a nivel de archivo.

Uso: Principal en macOS y iOS.

Btrfs (Linux)

Características: Snapshots, subvolúmenes, autocorrección de errores.

Uso: Algunas distribuciones de Linux.

15) **¿Qué es un deadlock y cómo puede prevenirse en un sistema operativo?**

Un deadlock es una situación en la que dos o más procesos quedan bloqueados indefinidamente, esperando recursos que están siendo utilizados por otros procesos en la misma situación.

Prevención de Deadlocks:

1. Evitar Exclusión Mutua: No siempre es posible, pero se busca compartir recursos cuando es viable.
2. Preempción de Recursos: Forzar la liberación de recursos si es necesario.
3. Evitar Espera Circular: Establecer un orden en el que los recursos pueden ser solicitados.

Detección y Recuperación:

- Detección: Monitorear el sistema para identificar deadlocks.
- Recuperación: Terminar procesos involucrados o forzar liberación de recursos.