

1. Componentes y tipos de sistema operativos

- i. ¿Cuáles son las tres funciones principales del sistema operativo?
- ii. ¿Qué diferencias en cuanto a diseño y objetivos puede existir entre un sistema operativo de una máquina personal (laptop) al de un sistema Mainframe?
- iii. ¿Qué diferencias en cuanto a diseño y objetivos puede existir entre un sistema operativo móvil y uno de una máquina personal laptop?
- iv. ¿Qué motivos podría argumentar a favor de incluir los navegadores de internet dentro del sistema operativo? ¿Qué motivos en contra?
- v. Explique a que se refiere el modo de Kernel ¿Puede ayudar a proteger a la computadora contra uso indebido? ¿Cómo?
- vi. ¿Puede una máquina virtual tener modo supervisor? Si es afirmativo entonces: ¿Cómo es que hace el sistema operativo para que la MV pueda ejecutar instrucciones en modo supervisor sin comprometer la seguridad de la máquina?
- vii. ¿En qué se diferencia un sistema cliente servidor de uno P2P (peer to peer)?
- viii. ¿En qué se diferencia un sistema multiprocesador a un sistema cluster?
- ix. Describa el mecanismo de funcionamiento de una interrupción ¿En qué circunstancias es que este mecanismo es necesario?
- x. ¿Es posible crear un sistema operativo seguro en una máquina cuyo procesador no tenga modo seguro/supervisor?
- xi. En un sistema SMP, que ventajas/desventajas conlleva utilizar utilizar un único cache para todos los procesadores contra un cache para cada procesador
- xii. ¿Qué es el problema de la coherencia del cache? De un ejemplo de cómo se manifiesta
- xiii. ¿Qué ventajas/desventajas Ofrece el OpenSource para los sistemas operativos?

2. Llamadas al sistema

a. Llamadas por interrupción

- i. ¿Qué función cumplen las llamadas al sistema operativo? ¿Cuáles son las 5 categorías principales?
- ii. ¿Qué responsabilidades tiene el Sistema Operativo con respecto al manejo de memoria?
- iii. ¿Qué responsabilidades tiene el sistema operativo con respecto a almacenamiento secundario?
- iv. ¿En qué situaciones es preferible utilizar un interfaz gráfico? ¿En qué circunstancias es preferible un interfaz de línea de comandos?
- v. ¿A qué se refieren cuando hablan de un SO diseñado en capas?
- vi. ¿Cómo es que se lleva a cabo el proceso de inicio del sistema operativo en el momento en que la máquina es encendida?

- vii. ¿Qué ventajas/desventajas tiene el usar el mismo interfaz para manejar dispositivos que para manejar archivos?
- viii. ¿Qué es un sistema operativo de microkernel? ¿Qué otras opciones hay?
- ix. ¿Cuáles son las ventajas/desventajas de utilizar módulos cargables (loadable)?
- x. ¿Cuándo el sistema operativo carga un programa, cuales son los pasos que debe resolver el loader?
- xi. ¿Cuáles mecanismos de paso de parámetros existen entre los programas usuario y los system calls?

3. Administración de procesos

- i. Dado el siguiente programa


```
int main() {
    fork();
    int id = fork();
    if (id == 0) fork()
}
```

 ¿Cuántos procesos se generan en total?
- ii. ¿Cuáles son los tres aspectos o complicaciones que un sistema operativo debe cuidar cuando permite la ejecución de procesos concurrentes.
- b. PCB y el concepto de proceso
- c. Scheduling de procesos (ready, wait, blocked queue)
- d. Operaciones en procesos
- e. Comunicación entre procesos
 - i. Mensajes
 - ii. Memoria compartida (pros y cons)
- f. Ejemplos bibliotecas para la comunicación entre procesos (Posix, etc.)
- g. ¿Qué acciones toma el kernel cuando efectúa un cambio de contexto?
- h. Indique que sucede con un proceso desde que efectúa una instrucción de IO (ie lectura de un disco) hasta que esta termina de ser procesada y el proceso recupera el uso del CPU.

4. Threads

- a. (Falso/Verdadero) los registros del CPU se mantienen de un thread a otro
- b. (Falso/Verdadero) el Heap de memoria se mantiene de un thread a otro
- c. (Falso/Verdadero) las variables globales se mantienen de un thread a otro
- d. (Falso/Verdadero) la memoria del stack se mantiene de un thread a otro
- e. ¿Qué diferencias existen entre concurrencia y paralelismo?

5. Sincronización

- a. ¿En qué circunstancias es mejor usar un spinlock?
- b. ¿Cuáles son las tres condiciones necesarias para una región crítica?
- c. ¿Qué riesgos pueden suceder si deshabilito interrupciones para garantizar atomicidad en un sistema de un único procesador, si la des-habilitación se hace en un proceso de usuario.
- d. ¿En qué estructuras de datos del sistema operativo se podrían dar condiciones de carrera (race conditions)?

- e. De el código para implementar un lock mutex utilizando la instrucción de `test_and_set()`.
- f. De el código para implementar un lock mutex utilizando la instrucción de `compare_and_swap()`
- g. Utilizando `signal` y `wait`, escriba el pseudocódigo de un monitor que administre el acceso a un archivo del sistema (debe garantizar que el archivo sólo se le otorga acceso a un proceso a la vez). El monitor debe implementar el `getFile(proc)`, `retFile(proc)`, nada más. Puede asumir que `proc` es la identidad del proceso que está haciendo la solicitud.
- h. Demuestre como se puede implementar el `wait` y `signal` utilizando el `test_and_set()`

6. Sincronización y Scheduling

- a. Suponga que tiene

Proceso	Llegada	Tiempo de Burst
P1	0.0	7
P2	0.5	5
P3	0.9	3

¿Cual es el tiempo de turnaround para estos procesos utilizando FCFS? ¿SJF? ¿El tiempo de espera bajo las mismas condiciones?

- b. ¿Qué relación hay entre SJF y prioridad
- c. ¿Qué relación hay entre RR y SJF?
- d. ¿Cuáles de las políticas de asignación de CPU de FCFS, SJF, RR y Prioridad pueden conducir a starvation? ¿Porqué?