

Examen Parcial
Sistemas operativos
Primavera 2026

Sección Teórica

1.- ¿El modo usuario tiene acceso sin restricciones al hardware?

- a) Verdadero
- b) Falso

2.- ¿Qué virtualiza el sistema operativo?

- a) CPU
- b) MEMORIA
- c) DISCO DURO

3.- ¿Los procesos usan ____ para comunicar se con el kernel?

- a) System Calls
- b) Librerías
- c) Hilos

4.- ¿Tres componentes del sistema operativo para administrar la memoria, dispositivos, etc.?

- a) Virtualización, Persistencia, Concurrencia
- b) Procesos, Address Space, Stack
- c) Sistema de archivos, Hardware, Heap

5.- ¿Cuáles son los tres estados de un proceso?

- a) Running, Blocked, Ready
- b) Fork, Wait, Exec
- c) Joined, Detach, Yielded

6.- ¿Algoritmo que usa time-slices para simular paralelismo en procesos?

- a) Round-Robin
- b) FIFO
- b) Shortest Job First

7.- ¿El Address space de un proceso es compartido entre los hilos del mismo proceso?

- a) Verdadero
- b) Falso

8.- ¿Los procesos terminan voluntariamente cuando ____?

- a) Tiene salida por error
- b) Tiene un error fatal
- c) Por instrucción de un proceso padre

9.- ¿Sólo los programas en C tienen Stack y Heap?

- a) Verdadero
- b) Falso

10.- ¿Los primitivos son sólo parte del estándar de C y C++?

- a) Verdadero
- b) Falso

11.- ¿Los schedulers no tienen en cuenta la duración de los procesos por _____?

- a) Las interrupciones mismas del sistema no dejan determinar un tiempo definido
- b) Los procesos no saben su propia duración en el sistema
- c) El scheduler al hacer context switching no determina el tiempo de vida de un proceso

12.- ¿Los mutex sólo sirven para regular las race conditions?

- a) Verdadero
- b) Falso

13.- ¿Qué formas hay de comunicación entre hilos?

- a) Sockets, Condition variables, Memoria compartida
- b) Sockets, Condition Variables, Paso por mensaje
- c) Sockets, Atomic Variables, Promises

14.- ¿La función join() se puede llamar más de dos veces en un mismo hilo?

- a) Verdadero
- b) Falso

15.- ¿Qué es el user space?

El user space es la parte del sistema donde corren los programas normales que usamos. Ahí el código no tiene acceso directo al hardware ni a todo el sistema, sino que tiene que pedir permiso al kernel usando system calls, sería como la zona segura para que si un programa falla no se caiga todo el sistema operativo

16.-¿Cuáles son las partes del prompt de la terminal?

- Nombre de usuario
- Nombre de la máquina/equipo
- Directorio actual
- Símbolo de permiso: como \$ o # (usuario estándar o superusuario root)

Referencia teoría:

<https://pages.cs.wisc.edu/~remzi/OSTEP/cpu-intro.pdf>

<https://pages.cs.wisc.edu/~remzi/OSTEP/cpu-api.pdf>

<https://pages.cs.wisc.edu/~remzi/OSTEP/threads-intro.pdf>

<https://pages.cs.wisc.edu/~remzi/OSTEP/threads-api.pdf>

<https://pages.cs.wisc.edu/~remzi/OSTEP/threads-locks.pdf>

[f How Linux works](#) capítulo 1

<https://ebookcentral.up.elogim.com/lib/updf-ebooks/reader.action?docID=1848073&c=RVBVQg&ppg=11>

Sección Práctica

1.-¿Qué realizaría este comando?

Describirlo paso por paso

```
cd /etc; ls | grep cron
```

2.-¿Qué realizaría este comando?

Describirlo paso por paso

```
echo "alias code=codium" >> .bashrc
```

3.-¿Qué realizaría este comando?

Describirlo paso por paso

```
echo $PATH; export PATH=/usr/bin/vim:$PATH
```

4. Realizar un programa en C++ que escriba en un archivo de texto plano usando multithreading, cada thread tiene que escribir algo diferente y en el archivo donde se escribe se tiene que ver el texto de forma correcta, cada línea tiene que ser el texto de un thread diferente. Usen 5 threads diferentes para resolver el problema.

5. Ejecuten un one liner(una línea de un solo comando) que primero cree un archivo de texto plano llamado accept.txt, después de crearlo liste los contenidos del directorio y redirija su salida para escribir en este archivo de texto, después liste todos los comandos utilizados hasta ahora en su sesión de terminal, encuentre las veces que se repite en esta salida el comando ls y redirija la salida de ese comando al final

del archivo accept.txt

- Para entregar esta pregunta use el comando history al final de realizarla y copie y pegue la salida del comando en un archivo llamado pregunta5.sh agregue este archivo a su repositorio de examen

Formato de entrega:

Se entregará un documento en formato PDF y un link de repositorio con el contenido práctico del examen.

El nombre del archivo será en forma de

ID_Nombre_ApellidoPaterno_SO_ExamenParcial_1.pdf

El nombre del repositorio deberá ser **ExamenParcial1**.

Notas de evaluación:

Cualquier entrega fuera de plazo el examen se considerará anulado..

Cualquier entrega con el formato incorrecto el examen se considerará anulado.

Si el repositorio de código no es visible, o el profesor no puede acceder a él, el examen se considerará anulado