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

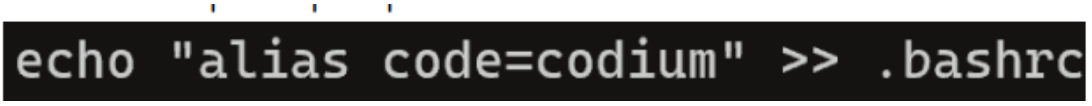
7.- ¿El Address space de un proceso es compartido entre los hilos del
mismo proceso?
a)Verdadero
b)Falso
8.- ¿Los procesos terminan voluntariamente cuando ____?
a)Tiene salida por error
b)Tiene un error fatal
c)Por instrucción de un proceso padre
9.- ¿Sólo los programas en C tienen Stack y Heap?
a)Verdadero
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? Es el espacio de memoria donde se corren los programas de usuario
16.-¿Cuáles son las partes del prompt de la terminal? Nombre de usuario, directorio actual, nombre del host
Referencia teoría:
https://pages.cs.wisc.edu/~remzi/OSTEP/cpu-intro.pdfhttps://pages.cs.wisc.edu/~remzi/OSTEP/cpu-api.pdfhttps://pages.cs.wisc.edu/~remzi/OSTEP/threads-intro.pdfhttps://pages.cs.wisc.edu/~remzi/OSTEP/threads-api.pdfhttps://pages.cs.wisc.edu/~remzi/OSTEP/threads-locks.pdf
How Linux works capitulo 1
https://ebookcentral.up.elogim.com/lib/updf-ebooks/reader.action?docl
D=1848073&c=RVBVQg&ppg=11

Sección Práctica
1.-¿Qué realizaría este comando?
Describirlo paso por paso

Te diriges al directorio etc y despues listas todas las coincidencias de files con la palabra cron

2.-¿Qué realizaría este comando?
Describirlo paso por paso



Estas metiendo la linea de texto “alias code=codium” a el archivo .bashrc, lo que crea un alias

3.-¿Qué realizaría este comando?
Describirlo paso por paso

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

Ves la ruta del path, despues la cambias a lo que dice el comando de export

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.

```
#include <iostream>
#include <fstream>
#include <thread>

void writeToFile(int id) {
    std::ofstream file("output.txt", std::ios::app);
    file << "Hilo " << id << std::endl;
}

int main() {
    std::thread t1(writeToFile, 1);
    std::thread t2(writeToFile, 2);
    std::thread t3(writeToFile, 3);
    std::thread t4(writeToFile, 4);
    std::thread t5(writeToFile, 5);
    t1.join();
    t2.join();
    t3.join();
    t4.join();
    t5.join();
    return 0;
}
```

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

```
touch accept.txt; ls | cat > accept.txt; history | grep "ls" | cat
```

```
>> accept.txt
```

- Para entregar esta pregunta use el commando 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