

NOMBRE: _____ MATRÍCULA: _____

1.- ¿Cuál de las siguientes es una aseveración falsa?

- a) Un Sist. Operativo es un programa que actua como intermediario entre el usuario y el hardware de la computadora.
 b) Un Sistema Operativo administra los recursos del sistema
 c) El Sistema Operativo controla la ejecución de programas de usuario

d) Los system calls son rutinas internas del Sist Op y no pueden ser utilizados por aplicaciones del usuario final.

2.- La principal contribución del Procesamiento Offline es:

- a) Traslape de I/O de un job con el cómputo de otros** c) Introducción de esquemas de protección
 b) Introducción de Vector de Interrupciones d) Introducción de Multiprogramación

3.- ¿ Cuál de las siguientes instrucciones debe ejecutarse en modo usuario en un sistema computacional que soporta los dos modos de ejecución de procesos ?

- a) Deshabilitar interrupciones **c) Obtener la raíz cuadrada de un número**
 b) Realizar una operación de escritura de información a disco d) Crear un nuevo proceso en el sistema

4.- Para la siguiente pregunta considere la tabla que muestra los jobs que se ejecutarán en un sistema con un solo procesador. En la tabla se muestra para cada job su número de indentificación, el tiempo de ejecución, y el tiempo de llegada.

Job ID	Tiempo de Ejecución	Tiempo de Llegada
1	2 seg	t
2	5 seg	t
3	1 seg	t + 1 seg
4	9 seg	t + 5 seg
5	3 seg	t + 10 seg
6	2 seg	t + 15 seg

Para una política de scheduling de round-robin con un cuantum de 10 milisegundos. El tiempo aproximado en el cual el job 3 termina es (NOTA: el cuantum esta en milisegundos y el tiempo de llegada, tiempo de ejecución y las opciones de respuesta son segundos):

- a) $t + 8$ seg **b) $t + 4$ seg** c) $t + 2$ seg d) $t + 1$ seg

5.- Para el siguiente problema asumir que el proceso A se encuentra en ejecución mientras ocurre una interrupción de I/O. Se requiere que indique la secuencia correcta en que los eventos que se listan deben de ocurrir

- A.- La rutina de atención a la interrupción termina su ejecución.
 B.- El control de la ejecución se retorna al proceso A.
 C.- Se transfiere control a la rutina de atención a la interrupción.
 D.- Se restaura el estado del proceso A.
 E.- El Sist.Operativo salva el estado (registros, program counter, etc.) del proceso A

- a) C, A, D, B, E b) C, A, D, E, B **c) E, C, A, D, B** d) E, C, A, B, D

6.- La computadora CDC 6600 maneja una variante de la política Round Robin conocida como *processor sharing*. En esta política hasta 10 procesos se manejan simultáneamente. Un cambio de contexto ocurre después de cada instrucción, de tal manera que la primer instrucción que se ejecuta proviene del proceso 1, la segunda instrucción del proceso 2, y el cambio de contexto entre instrucciones de diferentes procesos independientes toma " p " unidades de tiempo. Por ejemplo, si hay más de 10 procesos listos para ejecutarse se toman los 10 primeros y se ejecuta una instrucción de cada proceso con un overhead total de $10p$. Al concluir la ejecución de la primera instrucción de los 10 primeros procesos se cargan los siguientes 10 procesos para ejecutar la primera instrucción de estos y así sucesivamente. El overhead de cargar 10 procesos es de " k " unidades de tiempo. Si un proceso A requiere de " T " instrucciones de CPU (una instrucción representa una unidad de tiempo) para completar su ejecución cuando este es el único proceso en el sistema, ¿ Cuánto tiempo se requerirá para completar la ejecución del proceso A bajo la política de processor sharing cuando se tienen n ($n > 10$ y múltiplo de 10) procesos listos para ejecutarse en el sistema?. Asuma que el tiempo de CPU requerido por cada proceso es mayor o igual a " T " y que el proceso A es el n -ésimo proceso listo para ejecutarse.

- a) $nTk/10$ **b) $nT + nTk/10 + nTp$** c) $nT + nTk/10 + np$ d) $nTp + nTk/10$

7.- Se cuenta con un sistema de respaldos incremental el cual realiza los respaldos diariamente a las 2am. Considera una calendarización de los respaldos como la que se muestra en la tabla y considera las actividades que se realizan sobre archivos (en donde w.x = escribir archivo x, r.x = leer archivo x, d.x = borrar archivo x).

Hora	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
2am	R0	R1	R2	R2	R1	R2	R2
3am							
4am	r.a	r.a	w.a	w.a	w.a	r.d	
...	r.b	w.b	w.b	r.d	d.a	w.d	
...	w.c	w.d	w.d	w.b			
...							
12am							
1am							

Indique los archivos que son respaldados el día Jueves una vez que el respaldo de las 2am concluyó.

- a) a, b **b) a, b, d** c) a, b, c d) a, b, c, d

8.- Una persona ha abierto tres sesiones a un servidor UNIX-like . Asuma que cada sesión sólo tiene un proceso llamado bash. ¿El identificador del proceso es el mismo para las tres sesiones?

- a) Verdadero
- b) Falso**

9.- ¿Qué es una llamada a sistema (system call)?

- c) Es un proceso incluido en el sistema operativo.
- d) Es un API entre el usuario y el núcleo (kernel).
- e) Es la interfase entre un proceso y el sistema operativo. Ó**
- f) Es la interfase entre un proceso y el núcleo (kernel).**

10.- De acuerdo al libro de texto un proceso tiene cinco estados. Relacione los estados con la descripción de este estado.

Estado	Descripción
Listo (ready) 10	1. Terminó de ejecutarse pero está esperando que un dispositivo le entregue la información necesaria.
Espera (waiting) 5	<u>2.(Otro)</u>
Terminado (terminated) 8	3. Tiene la mejor calificación de todos los procesos.
Corriendo (running) 9	4. Terminó de ejecutarse pero no ha sido removido del sistema operativo.
Nuevo (new) 6	5. Está esperando a que ocurra un evento.
Bloqueado (Blocked) 1	6. Espera para ser admitido por primera vez en estado de listo.
Zombi (Zombie) 4	7. Es el que ocupará el procesador aunque no sea su turno 8. Su estado anterior es corriendo y no pasa al estado de listo o espera. 9. Proceso que está actualmente ejecutándose en al menos un procesador. 10. Aguarda a que se le asigne el procesador.

11.- Dado el archivo so1.txt que se muestra en el siguiente párrafo. ¿Cuál la salida del siguiente comando:?

Host # cat so1.txt so1.txt so1.txt

so1.txt

1,2,3,a,b

a) 1,2,3,a,b 1,2,3,a,b 1,2,3,a,b

c) 1,2,3,a,b

1,2,3,a,b

1,2,3,a,b

b) 1,2,3,a,b 1,2,3,a,b 1,2,3,a,b,1,2,3,a,b

d) ninguno de los anteriores

12. Dado el archivo so.txt que se muestra en el siguiente párrafo. ¿Cuál la salida del siguiente comando:?

Host# cat so.txt so.txt so.txt | head -n 3

so.txt

Sistemas operativos.

Me gustan los threads y los fork :)

- a) Sistemas operativos.
Me gustan los threads y los fork :)
Sistemas operativos.
Me gustan los threads y los fork :)
Sistemas operativos.
Me gustan los threads y los fork :)
- b) Sistemas operativos.**
Me gustan los threads y los fork :)
Sistemas operativos.

- c) Me gustan los threads y los fork :)
Me gustan los threads y los fork :)
Me gustan los threads y los fork :)

d) ninguno de los anteriores

13. Dado el código que se muestra a continuación. Llene los espacios con la palabras padre o hijo segun corresponda. ¿Cuál es la salida del código?

```
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
```

```

int main()
{
    pid_t pid;
    char *message;
    int n;

    printf("fork program starting\n");
    pid = fork();
    switch(pid)
    {
    case -1:
        perror("fork failed");
        exit(1);
    case 0:
        message = "This is the _____";
        n = 5;
        break;
    default:
        message = "This is the _____";
        n = 3;
        break;
    }

    for(; n > 0; n--) {
        puts(message);
        sleep(1);
    }
    exit(0);
}

```

14.- Dado el siguiente código describa la salida esperada. ¿Cuántos procesos genera este programa? ¿Por qué?

```

#include <unistd.h>
#include <stdio.h>
int main()
{
    printf("Running ps with execp\n");
    execp("ps", "ps", "-ax", 0);
    printf("Done.\n");
    exit(0);
}

```

15.- Si tengo dos sesiones. En una corro un programa mostrado a continuación y en el otro corro el comando kill

- a) ¿Cuál debe ser la sintaxis del comando para simular un ctrl+c?
- b) ¿Cuál va a ser la salida al enviar el primer kill?
- c) ¿Cuántas veces puede este código recibir ctrl+c?

```

#include <signal.h>
#include <stdio.h>
#include <unistd.h>
void ouch(int sig)
{
    printf("OUCH! - I got signal %d\n", sig);
    (void) signal(SIGINT, SIG_DFL);
}

int main()
{
    (void) signal(SIGINT, ouch);

    while(1) {
        printf("Hello World!\n");
        sleep(1);
    }
}

```