

FACULTAD DE INGENIERIA EN ELECTRICIDAD Y COMPUTACION
SISTEMAS OPERATIVOS
MEJORAMIENTO – II TERMINO 2019 - 2020

Nombre: _____ Matrícula: _____

Reglas: Escriba las respuestas con pluma, no lápiz. Justifique de forma suficiente cada respuesta, poner una sola oración no es una respuesta válida. Sea concreto y utilice los términos técnicos adecuados. Puede colocar gráficos ser necesario en hoja auxiliar.

PREGUNTAS

1) ¿Por qué el manejo de un fallo de página es más complejo que el manejo de una interrupción de software o hardware? (10 pts)

2) ¿Cuál es la diferencia de entre un planificador apropiativo y uno no-apropiativo? ¿Alguno requiere soporte de hardware?(10 pts)

3) ¿Si se aplica el método de copy-on-write que sección(es) del proceso (texto, datos, stack) es más probable de permanecer en páginas compartidas y por qué?. (10 pts)

4) ¿Que es un Shell? ¿Describe como explota las llamadas al sistema para que se ejecuten tareas? (10 pts)

5) ¿En términos de uso eficiente del CPU, cuando es recomendable y cuando no usar spin locks (10 pts)

6) ¿Bajo qué condiciones la apropiación de recursos como mecanismo para evitar interbloqueo podría funcionar? (10 pts)

PROBLEMAS

7) Considere un proceso con la mitad de su tabla de páginas está cargada en la TLB, si el tiempo de acceso a la TLB es 50ns y a la RAM es 50 ms. Cuál será el tiempo de acceso efectivo si se asume que hay la misma probabilidad de acceder a cualquier página de memoria virtual. (5 pts)

8) El software de manejo de versiones SUBersion debe de funcionar así: solo pueden actualizar sus programas hasta 4 clientes remotos, si hay menos de 4 clientes remotos un nuevo cliente puede actualizar su versión del programa, si ya hay 4 clientes, los nuevos clientes deben esperar a que TODOS los 4 clientes hayan terminado. Implemente en pseudocódigo la solución al problema de la sección crítica. (10 pts)

9) Considere el siguiente string de referencia: 1,3,4,5,3,6,1,3,7,8,3,2 ¿Cuántos fallos de página ocurrirían, asumiendo 4 frames para los algoritmos *LRU* y *Optimal*? (14 pts)

10) El siguiente programa usa el API pthread

<pre>int value = 0; void *runner(void *param); /* the thread */ int main(int argc, char *argv[]) { int pid; pthread_t tid; pthread_attr_t attr; pid = fork(); if (0 == pid) { pthread_attr_init(&attr); pthread_create(&tid, &attr, runner, NULL); pthread_join(tid, NULL); printf(" value = %d\n", value); /* LINE C */ } else if (0 < pid) { /* Parent process */ wait(NULL); printf("value = %d\n", value); /* LINE P */ } }</pre>	<pre>void *runner(void *param) { value = 5; pthread_exit(0); }</pre> <p>a. ¿Qué imprime la línea C? (3 pts)</p> <p>b. ¿Qué imprime la línea P? (3 pts)</p>
--	--

11) Dada la siguiente tabla de procesos:

Process ID	Arrival Time	Burst Time
P1	0	3
P2	1	3
P3	2	1
P4	3	4

Si se emplea el algoritmo FCFS y se sabe que hay un overhead al planificar cada proceso, equivalente a 1 unidad de tiempo. Calcule:

- a) El diagrama de Gantt (5 pts)
- b) Cual es el tiempo perdido en overhead? (2.5 pts)
- c) Cual es el tiempo empleado en trabajo útil del CPU? (2.5 pts)