**Sistemas Operativos.**

Segundo Examen Parcial ***Enero – Mayo 2020.***

Nombre: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Clave Única: \_\_\_\_\_\_\_\_\_\_\_ Calificación: \_\_\_\_\_\_\_

Entre [ ] aparecen los puntos de cada problema. **Conteste usando TINTA Negra. Duración: 1:50 hs.**

1)**[2.2]** Programa concurrente.

1. ¿Defina qué es un programa concurrente?
2. ¿Cómo se ve en la memoria, un programa concurrente en ejecución? Soporte su explicación en un dibujo.
3. De dos razones para tener procesos concurrente cooperantes, como resultado de haber dividido un único proceso grande.
4. Se tienen dos archivos *Uno.cc* y *Dos.cc*. El archivo *Uno.cc* contiene la instrucción *system(“Dos.exe”)* donde *Dos.exe* es el ejecutable de *Dos.cc*. Muestre, en ejecución el árbol jerárquico de procesos a partir de mandar ejecutar *Uno.exe*.

2)**[2.2]** Algoritmos de planificación (algorithm scheduling).

1. De los dos grandes grupos en que se dividen los algoritmos de planificación, indique dos diferencias de funcionamiento entre estos grandes grupos, una diferencia desde el punto de vista de los estados READY y RUNNING; y la otra diferencia desde el punto de vista de que pasa con los *CPU burst*s en uno y otro grupo de algoritmos.
2. ¿Qué problemas presenta, para el sistema, el algoritmo RR, Round Robin, cuando su quantum o rebanada es muy grande?
3. Explique en qué consiste el *cambio de contexto*. ¿En cuántos casos posibles se da cambio de contexto? Explíquelos en función de los estados del Time Sharing.
4. En cuál de los cuatro algoritmos de planificación, vistos en clase, se aplica el concepto de *prioridad* a los procesos. ¿En qué consiste este concepto, qué importancia tiene para el funcionamiento del scheduling y en cuál estado se genera?

3) **[1.2]** Suponga un Sistema de Cómputo Multiprogramable que atiende las I/O, de los procesos, por poleo.

1. Dibuje el diagrama de estados, con los cinco estados para este caso.
2. Explique con detalle el funcionamiento entre estados.

**>>> SIGUIENTE PÁGINA**

4) **[2.2]** Analice el siguiente programa X.cc.

|  |
| --- |
| #include . . .  int main(int argn, char \*argv[])  {  int i;  for(i=0; i<4; i++)  {  fork();  printf(“i=%d\n”,i);  }  return 0;  } |

y conteste:

1. Dibuje el diagrama del árbol de procesos creado a partir del proceso *X*., e incluya, en cada proceso, el valor de i (“i=”) que cada proceso imprime.

**b)** ¿Cuántos procesos en total fueron creados a partir del proceso X.

5)**[2.2]** Llene en el archivo “SRTF2ex.xls”, que se adjunta, las diferentes gráficas de Gant (cola de Ready, Running, colas de Waiting y Terminated), según corresponda, para el caso de cuatro procesos, que son planificados por el algoritmo preemptivo de planificación SRTF, mostrándose sus tiempos estimados, tiempo de arribo, el tiempo de los dos burst reales, así como el único burst de IO. Esto quiere decir que cada proceso tiene dos bursts de CPU y un burst de IO antes de terminar. Los dos burst reales de CPU, de cada proceso, son iguales.

Usted sólo deberá llenar las gráficas de Gant hasta que vayan terminando los procesos y / o hasta que se le acaben los cuadros de las gráficas de Gant.

Tal y como desarrollamos los ejercicios de esta naturaleza en clase, las gráficas deben incluir los tiempos. Seguiremos considerando al valor “alfa” como ½.

**<<< PÁGINA ANTERIOR**