UTN FRD SO – Segundo Examen Parcial Tema I

1) File System. Dados los siguientes parámetros:
nroPrimerRF: Número del bloque físico del disco donde comienza un archivo.
FB : cantidad de registros lógicos que almacena un bloque o RF del disco.
nroRegistroLogico: Número de un registro lógico relativo a 0 (cero), que corresponde con el órden lógico del registro lógico dentro del archivo.
nroRFcontenedorRL: Número del bloque o registro físico del disco que contiene al registro lógico.
posRLdentroRF: Orden en el cual se encuentra el registro lógico dentro del bloque o registro físico que lo contiene.

a) Escriba la expresión algebraica para obtener nroRFcontenedorRL y posRLdentroRF en asignación contigua:

nroRFcontenedorRL = expresión algebraica
posRLdentroRF = expresión algebraica

b) Escriba una función que retorne nroRFcontenedorRL en asignación enlazada
nroRFcontenedorRL funcion (.....) //puede escribirla en pseudocodigo

c) Escriba una función que retorne nroRFcontenedorRL en asignación indexada
nroRFcontenedorRL funcion (.....) //puede escribirla en pseudocodigo

2) Memoria. Dado el siguiente programa assembler, en un sistema de paginación bajo demanda, donde las direcciones lógicas de memoria son de 16 bits y la página tiene una capacidad de almacenamiento de 4 KBytes y los números de frames libres son los siguientes 7, A, D

Proceso en Memoria Virtual	**Responda:**

`mov AX,[412A]`
`mov BX,[245B]`
`mov CX,[378C]`
`add AX,BX`
`mov [612F],AX`
`mov AX,[A12A]`
`add BX,[B567]`
`mov [1234],BX`
	1) Construya la tabla de páginas
	2) Por cada fallo de página y utilizando el algoritmos LRU, actualice la tabla de páginas
	3) por cada acceso a memoria a datos mapee la dirección lógica a física.

3) Planificación de CPU. Planificación de CPU próxima ráfaga más corta primero con desalojo. Dados dos procesos con la siguiente definición:
Proceso A: 20msCPU, 30msDISCO, 30msCPU, 30msDISCO, 10msCPU
Proceso B: 30msCPU, 20msDISCO, 20msCPU, 30msDISCO, 20msCPU
Pronóstico para la primera ráfaga de CPU de A y B: 10ms. T0=0
1) Grilla de ejecución SO, A, B, interrupciones, indicador de rutinas según pizarrón, cola de listos
2) Tiempo medio de retorno
3) Porcentaje de uso de CPU
4) Porcentaje de sobrecarga del SO
5) Tiempo medio de espera.

4) Un proceso productor usa un buffer circular en memoria compartida (0xA, ya está creada) de 320 bytes, cada producción ocupa 32bytes. Existe un único proceso consumidor. La función void produce(char *to) produce 32bytes en formato string (incluye \0) y lo copia en to, el consumidor solo imprime en pantalla el string consumido. Todo finaliza cuando se produce el mensaje "chau". Debe indicar los semáforos que necesita, sus valores iniciales, se asume que los mismos están creados e inicializados, cuenta con las funciones SemWait(semid,nro sem) y SemSignal(semid,nro sem) solo **complete el código principal del proceso consumidor**. Le damos como ayuda una parte **incompleta y errónea** del código del productor:

```
int shmid = shmget(0xA,0,0), salir=0;
char *p = (char *) shmat(shmid,0,0);
do {     produce(p);
     if ( strncmp(p,"chau",4) == 0 ) salir=1;
     p+=32;
} while(!salir);
```