

# Sistemas Operativos - OSTEP

## VIRTUALIZACIÓN

### Ejercicio 1.

- (a) Realizar el diagrama de transición de estados de un proceso.
- (b) Indicar cual es la transición que posibilita los planificadores apropiativos como *round robin*.
- (c) Explicar que es un proceso *zombie*.

### Ejercicio 2. Para el siguiente código

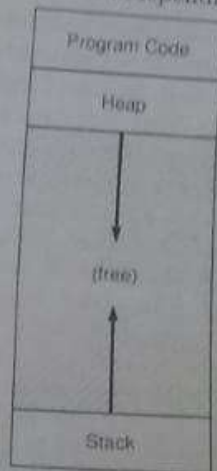
- (a) Indicar en que parte de la memoria está el contenido cada una de las variables. Si es un puntero indicar también a donde apunta.
- (b) Si lanzo **dos procesos** con este mismo código ¿Qué se comparte y qué es independiente?
- (c) Si lanzo **dos hilos** con este mismo código ¿Qué se comparte y qué es independiente?

```

int a[N];

int main(int argc, char ** argv)
{
    int i;
    register int s = 0;
    int *b = calloc(N, sizeof(int));
    for (i=0; i<N; ++i)
        s += a[i]+b[i];

    free(b);
    return s;
}
    
```



Ejercicio 3. Se quiere mitigar un problema de seguridad que permite leer las páginas físicas. La estrategia es intercambiar las páginas pares con las impares en todo el mapa de memoria física de 8 MiB a fin de entorpecer la lectura de la información. El sistema de paginado es de i386: direcciones virtuales de 32 bits, direcciones físicas de 32 bits, 10 bits de índice de *page directory*, 10 bits de índice de *table directory*, y 12 bits de *offset* dentro de la página. Mostrar el directorio, las tablas de página y dentro de estas cada uno de los 20 o 5 dígitos hexadecimales de la página física.

## CONCURRENCIA

Ejercicio 4. Considere el siguiente multiprograma de dos componentes P0 y P1, donde la atomicidad es línea a línea y todas las variables  $(i, j, a)$  son compartidas.

Pre: $i=0 \wedge j=0$	
P0 : while( $i < 32$ ) { $a[i] = j$ ; $++i$ ; }	P1 : while( $j < 32$ ) { $++j$ ; }
Post: ?	

- (a) Expresar de manera concisa todos resultados posibles en el arreglo  $a[0, 32)$ . Explique.
- (b) ¿Se puede cumplir con la postcondición  $(\forall k : 1 \leq k < 32 : a[k-1] < a[k])$ ? Explique.
- (c) Sincronice con semáforos para lograr que siempre se satisfaga la postcondición  $(\forall k : 1 \leq k < 32 : a[k-1] < a[k])$  sin hacer peligrar la terminación y maximizando la concurrencia.

Ejercicio 5. La Figura 1 muestra solución incorrecta al problema de la sección crítica de dos procesos, presentada por Hyman en 1966, a fin de competir con el algoritmo de Dekker.

- (a) Muestre un escenario de ejecución donde no se cumple la condición de sección crítica.
- (b) ¿Siempre funcional mal? Explique.

```

Pre : ¬flag0 ∧ ¬flag1 ∧ turn = 0
P0 : do true →
1   NCS0
2   ; flag0 := true
3   ; do turn = 1 →
4     do flag1 → skip
5     od
6     ;turn := 0
7   od
8   ; CS0
9   ; flag0 := false
   od
P1 : do true →
A   NCS1
B   ; flag1 := true
C   ; do turn = 0 →
D     do flag0 → skip
E     od
F     ;turn := 1
G   od
H   ; CS1
I   ; flag1 := false
   od
  
```

Figura 1: Hyman

Ejercicio 6. Enumere y explique las cuatro condiciones necesarias para que se produzca deadlock.

PERSISTENCIA

Ejercicio 7. El disco Western Digital Red de 4 TiB e interfaz SATA tiene una velocidad de rotación de 5400 RPM, 5.6ms de latencia de búsqueda y 150 MiB/s de tasa de transferencia máxima.

- (a) Indicar cuantos ms tarda en dar una vuelta completa.
- (b) Indicar la tasa de transferencia de lectura al azar de bloques de 4096 bytes.
- (c) Si la tasa de transferencia máxima está dada por la velocidad rotacional, calcular cuantos MiB tiene una pista (track).

Ejercicio 8. El siguiente es un sistema de archivos FAT un tanto corrupto, que se compone de un rootdir (Figura 2) y una tabla FAT como sigue, donde el cluster es de 512 bytes, se numeran desde 1, el 0 marca un cluster libre y la E indica fin de cadena.

- (a) Enumere y explique brevemente los cinco problemas esta estructura FAT.
- (b) Corrija al menos dos, explicando claramente en cada uno porque y como modificaría el rootdir y la FAT.

Nombre	Tamaño	Inicial
command.com	1200	1
smartdrv.sys	1024	4
autoexec.bat	524	5
config.sys	131	7
loom.exe	3000	9
tentacle.exe	3001	13

Figura 2: ROOTDIR

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
1	21	31	E1	61	E1	81	71	E1	101	111	121	141	151	161	171	E1	181	141	01	01	221	E1	131

Ejercicio 9. Supongamos que se borraron los dos mapa de bits con los bloques en uso y los i-nodos en uso de un filesystem tipo UNIX. Explique el procedimiento para reconstruir ambos.