

# Sistemas Operativos

## VIRTUALIZACIÓN

Ejercicio 1. Explique detalladamente como funcionan estos programas indicando cuantos procesos se crean, que hacen los hijos y padres, que archivos se cierran y abren, etc.

Suponga que en ambos casos se invocan con

```
$ ./a.out A B C D E F
```

```
int main(int argc, char ** argv) {
    int rc = fork();
    if (rc<0)
        return -1;
    else if (0==rc)
        return 0;
    else {
        execvp(argv[0], argv);
    }
}
```

```
int main(int argc, char ** argv) {
    char buf[L] = {'\0'};
    if (argc<=1)
        return 0;
    int rc = fork();
    if (rc<0)
        return -1;
    if (0==rc) {
        close(0);
        open(argv[0], 0);
        read(0, buf, L);
        write(1, buf, L);
    } else {
        argv[argc-1] = NULL;
        execvp(argv[0], argv);
    }
}
```

Ejercicio 2. Para el sistema de paginado i386 (10, 10, 12), sabiendo que CR3=0x00000 y que el contenido de los marcos físicos son los siguientes:

0x00000	0x10000
-----	-----
0x000: 0x10000 (P)	0x000: 0x00000 (P)
0x001: 0x10000 (P)	0x001: 0x00001 (P)
...	...
...	...
0x3FE: 0x10000 (P)	0x3FE: 0x003FE (P)
0x3FF: 0x00000 (P)	0x3FF: 0x003FF (P)

- Pasar de virtual a física: 0x00000EEF, 0x00001FEE, 0x00400EEF, 0x00401FEE.
- Indique qué contiene la dirección de memoria virtual 0xFFC00CFF.
- Modificar el mapa de memoria virtual de forma tal que 0x00000EEF apunte a 0x00000EEF y 0x00400EEF apunte a 0x00001EEF.

Ejercicio 3. Para el programa de la Figura 1, donde la **atomicidad es línea-a-línea**:

- (a) Muestre un (1) escenario de ejecución con  $N = 5$  de forma tal que la salida del multiprograma cumpla con  $a = [0, 1, 0, 1, 0]$ .
- (b) Muestre un (1) escenario de ejecución con  $N = 4$  de forma tal que la salida del multiprograma cumpla con  $a = [0, 0, 2, 2]$ .
- (c) Exprese **todos los valores posibles** de salida de arreglo  $a$  para la Figura 2. Explique.
- (d) Sincronice el multiprograma de la Figura 1 **usando semáforos** para que el resultado **siempre** sea  $a = [0, 1, \dots, 0, 1]$  para cualquier valor de  $0 < N$ .

Pre: $0 < N \wedge i, j = 0 \wedge (\forall k : 0 \leq k < N : a[k] = 2)$	
1 P0: while (i < N) { 2     a[i] = 0; 3     ++i; }	a P1: while (j < N) { b     a[j] = 1; c     ++j; }
a=?	

Figura 1: Concurrent Vector Writing (CVW)

Pre: $0 < N \wedge i, j = 0 \wedge s, t = 0, 1 \wedge (\forall k : 0 \leq k < N : a[k] = 2)$	
P0: while (i < N) { sem_wait(s); a[i] = 0; ++i; sem_post(t); }	P1: while (j < N) { sem_wait(t); a[j] = 1; ++j; sem_post(s); }
a=?	

Figura 2: CVW sincronizado

Ejercicio 4. Considere los procesos P0 y P1 a continuación, donde las sentencias son atómicas.

Pre: $n = 1$	
P0 : while (n < 100) { n = n*2; }	P1 : while (n < 100) { n = n*3; }
Post: $n = 192$	

- (a) ¿Se cumple la postcondición? Demostración rigurosa o contraejemplo.
- (b) Sincronizar con semáforos para que siempre dé como resultado  $n=125$ . Puede colocar condicionales (if) sobre el valor de  $n$  para hacer wait/post de los semáforos.
- (c) Sincronizar con semáforos para que siempre dé como resultado  $n=162$ . Puede colocar condicionales (if) sobre el valor de  $n$  para hacer wait/post de los semáforos.

Ejercicio 5. El disco *Pepito Digital Green* de 2 TiB e interfaz SATA-3 tiene una velocidad de rotacional de 4500 RPM, 6.5 ms de latencia de búsqueda y 80 MiB/s de tasa de transferencia máxima (un disco rotacional para el olvido).

- (a) Indicar cuantos *ms* tarda en dar una vuelta completa.
- (b) Indicar la tasa de transferencia de lectura **al azar** de bloques de 8 MiB.
- (c) Si la tasa de transferencia máxima está dada por la velocidad rotacional que no requiere cambio de pista (no sufre del *seek time*), deducir cuantos MiB almacena cada pista.

Ejercicio 6. En un sistema de archivos de tipo UNIX, tenemos los bloques de disco dispuestos dentro del *i-nodo* con 8 bloques directos, 1 bloque indirecto y 1 bloque doble indirecto. Cada bloque es de 1 KiB y los números de bloque ocupan 16 bits.

- (a) Calcule la capacidad máxima de un archivo.
- (b) Calcule la capacidad máxima del *block pool* del sistema de archivos.  
¿Hay algo raro entre la capacidad máxima de un archivo y del disco?
- (c) ¿A partir de que tamaño en KiB el archivo empieza a utilizar los bloques doble indirectos?