Exámen Final – 7 de diciembre 2021 Nombre: .....

Sistemas Operativos 2021 – OSTEP

Virtualización

Ejercicio 1. Explique detalladamente como funcionan estos programas. Suponga que en ambos casos se invocan con

```
$ ./a.out ./a.out ./a.out ./a.out ./a.out ./a.out ./a.out
```

```c
int main(int argc, char ** argv) {
  argc--;
  if (0<argc) {
    argv++;
    execvp(argv[0], argv);
  }

  return 0;
}
```

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

Ejercicio 2. Mostrar la secuencia de accesos a la memoria física que se produce al ejecutar este programa assembler `x86_32`, donde $Base_{code}$ = `0x1000`, $Bounds_{code}$ = `0x100`, $Base_{data}$ = `0x2000`, $Bounds_{data}$ = `0x10`
La versión original de este código salió en el número de abril de 1967 de Cosmopolitan.

```
  0: mov   $0x0,%edx
  6: mov   0x0(%edx),%ecx
 12: mov   0x4(%edx),%eax
 18: add   $0x8,%edx
 21: add   (%edx),%ecx
 23: add   $0x4,%edx
 26: sub   $0x1,%eax
 29: jne   21
```

```
0x0:  .long 0x0
0x4:  .long 0x4
0x8:  .long 0x3
0xC:  .long 0x7
0x10: .long 0x2
0x14: .long 0x1
0x18: .long 0x5
```

Ejercicio 3. Para el sistema de paginado i386 $(10, 10, 12)$, sabiendo que `CR3=0x0C01A` y que el contenido de los marcos físicos son los siguientes:

```
0x0C0CA                0x0C01A
------------------     ------------------
0x000: 0x0C01A (P)     0x000: 0x0C0CA (P)
0x001: 0x0C0CA (P)     0x001: 0x0C01A (P)
...                    ...
...                    ...
0x3FE: 0x0C01A (P)     0x3FE: 0x0C0CA (P)
0x3FF: 0x0C0CA (P)     0x3FF: 0x0C01A (P)
```

(a) Pasar de virtual a física: `0x00000BAD`, `0x00001BE1`, `0x00400BA1`, `0x00401BE5`.

(b) Pasar de física a **todas** las virtuales: `0x0C0CABAD`, `0x0C01ABE5`, `0xDEADBEEF`, `0xDABBAD00`.

Ejercicio 4. Considere el siguiente multiprograma de 3 componentes, donde `x` es compartida y **el incremento NO es atómico**. Exprese de manera concisa todos resultados posibles de la variable compartida `x` para cada uno de estos valores iniciales de semáforo `s={0,1,2,3}`

Pre: x=0 $\wedge$ s=?		

`P0 : down(s);`
`x = x+1;`
`up(s);`
	`P1 : down(s);`
	`x = x+1;`
	`up(s);`
		`P2 : down(s);`
		`x = x+1;`
		`up(s);`

Post: x =?		

Ejercicio 5. 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.

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 6. El disco *Western Digital Red NAS* de 4 TiB e intefaz SATA tiene una velocidad de rotacional 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 16 MiB.

(c) ¿La tasa de transferencia al azar de bloques de 128 MiB será mayor o menor a la anterior? Calcule y/o explique.

(d) 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 7. En un sistema de archivos de tipo UNIX, tenemos los bloques de disco dispuestos dentro del *i-nodo* con 12 bloques directos, 1 bloque indirecto y 1 bloque doble indirecto. Cada bloque es de 4 KiB y los números de bloque ocupan 24 bits.

(a) Calcule la capacidad máxima de la *data region* del sistema de archivos.

(b) Calcule la capacidad máxima de un archivo.

(c) Calcule para que tamaño se pasa de bloques directos a indirecto y para que tamaño de indirecto a doble indirecto.