

Entrega 6

Sistemas Operativos II

Mellino, Natalia

Farizano, Juan Ignacio

Ejercicio 1

- a) **Verdadero:** como se ve en el ejemplo del apunte de Señales `kill(getpid(), signalcode)` y `raise(signalcode)` son expresiones equivalentes. Con la primer operación estamos obteniendo el PID de nuestro propio proceso, y llamamos a `kill` con el código 'signalcode' y esto resulta en enviarnos a nosotros mismo una señal representada por el código mencionado, obteniendo el mismo resultado que en la llamada a `raise`.
- b) **Falso:** con `raise` un proceso sólo puede enviarse señales a sí mismo, en cambio, con `kill`, un proceso puede enviar señales a cualquier otro proceso del cual sepa su PID.
- c) **Falso:** `signal` no envía señales. Esta función toma una señal y un handler (función) como argumento. Cuando esa señal es interceptada, invoca al handler pasándole esa señal como argumento.
- d) **Verdadero:** Por definición, esta señal es enviada cuando ocurre una violación de segmento o referencia inválida a memoria virtual.

Ejercicio 2

Si el handler simplemente imprime un cartel, al producirse el fallo de segmentación se invocará al handler, se imprimirá el cartel, retornará y se volverá a la última instrucción ejecutada, es decir, volverá a suceder la violación de segmento y se invocará el handler repitiendo todo el proceso infinitamente.

Ejercicio 3

La diferencia principal es que el intercambio presentado en la sección 31.1.3 guarda en el disco una imagen del proceso entero mientras que en el intercambio parcial la memoria asignada al programa es subdividida en segmentos más pequeños y como durante la ejecución algunos de estos segmentos pueden no emplearse por largos períodos (o nunca), es posible enviarlos al disco.

Ejercicio 4

1	2	3	4	1	2	5	1	2	3	4	5
1	1	1	1	1	1	5	5	5	5	4	4
	2	2	2	2	2	2	1	1	1	1	5
		3	3	3	3	3	3	2	2	2	2
			4	4	4	4	4	4	3	3	3
M	M	M	M	H	H	M	M	M	M	M	M

2 aciertos (Misses)

10 fallos (Hits)