

Ejercicio de Sistemas Operativos

Parte Práctica

Problema 2

Responda a las siguientes preguntas relativas a los procesos y al API de llamadas al sistema POSIX relacionadas con procesos.

1. Dado el siguiente programa:

```

10 #include <stdlib.h>
11 #include <stdio.h>
12
13 int total;
14
15 int calcula(int i) {
16     if (i <= 2) {
17         total = total - (i*100);
18     }
19 }
20
21 int main() {
22     int i ;
23     int pid;
24
25     total=0;
26
27     for (i=0; i < 6 ; i++) {
28         pid=fork();
29         if (pid==0) {
30             calcula(i);
31             sleep(1000);
32         }
33     }
34     sleep(10);
35     printf ("El total es %d\n",total);
36 }
```

Asumiendo que el tiempo que tarda cada proceso en ejecutar su código es prácticamente despreciable, ¿se puede precisar qué valor mostraría el mensaje en la línea 35 en el instante aproximado $t = 10$? ¿En qué instantes de tiempo aproximadamente finalizarían cada uno de los procesos presentes en el ejercicio? Razone la respuesta.

2. Considere ahora el siguiente programa:

```

10 #include <stdlib.h>
11 #include <stdio.h>
12 #include <unistd.h>
13
14 int main() {
15     int i ;
16     int pid;
17
18     for (i=0; i < 6 ; i++) {
19         pid=fork();
20         if (pid==0) {
21             execl("/bin/ps", "ps", "-o", "pid,ppid,comm", (char*)0);
22         }
23     }
24     sleep(5);
25     printf("Fin del proceso %i\n",pid);
26 }
```

La ejecución de la orden `ps -o pid,ppid,comm` de la línea 21 muestra por pantalla un listado de todos los procesos lanzados por el usuario correspondiente. Dicho listado es una tabla con tres columnas indicando el PID, PPID y `COMMAND`¹ de cada uno de los procesos.

Sabiendo que se utiliza una *shell* `bash` para ejecutar el programa, que el PID de esa *shell* es 7463, que los PID en este sistema se asignan de forma correlativa, que el primer PID disponible es el 19602, y que no hay nadie más en el sistema creando procesos, ¿cuál sería la salida por pantalla del programa? Razone la respuesta.

Nota: Recuerde que en UNIX, tal y como está planteado el programa, los posibles procesos hijo creados no terminarán hasta que no lo haga su padre.

¹COMMAND es el nombre del archivo que contiene el código a ejecutar