Técnicas Digitales III

Trabajo práctico: Procesos

Creación de procesos

1. Compile y ejecute proc 01.c

Compile el programa \$ gcc -o proc_01 proc_01.c

Ejecute \$./proc_01

¿Qué es el PID ? ¿Cómo es el PID del padre respecto del PID del hijo?, ¿por qué?

2. Compile y ejecute proc_02.c. Luego, desde otra consola, ejecute el comando "pstree -p" e identifique los procesos en ejecución.

Compile el programa \$ gcc -o proc_02 proc_02.c

Ejecute \$./proc_02

En otra consola visualice el árbol de procesos con comando \$ pstree -p

- 3. El programa proc_03.c ejecuta 2 veces la función fork() y hace una espera activa de 30 segundos con la función sleep(). Compile y ejecute proc_03.c. Luego, desde otra consola ejecute el comando pstree -p e identifique los procesos en ejecución. ¿Cuántos procesos ejecutan la línea 23?
- 4. Tome proc_02.c y ponga las funciones de las líneas 14 y 15 dentro de un bucle que se repita 3 veces. Imprima también el valor de la variable de control del bucle (variable i). Analice y deduzca cuántos hijos son creados. Justifique su respuesta. ¿Qué sucede con el valor de i?.
- 5. La función fork() devuelve el pid del proceso hijo cuando lo ejecuta el padre; y devuelve 0 (cero) cuando lo ejecuta el proceso hijo. ¿Qué estructura de bifurcación de C le parece más conveniente para implementar que padre e hijo ejecuten diferente código (if, while, for, case)? Modifique proc_05.c con la estructura de bifurcación seleccionada. Compile y ejecute el programa.
- 6. Compile y ejecute proc_06.c. La variable x es inicializada en 100. Luego es decrementada por el proceso hijo e incrementada por el proceso padre. ¿Por qué x nunca retorna a su valor original?. Se imprime la dirección en memoria de la variable x (&x). ¿Cómo es el valor dirección, por qué?

Creación de procesos en Python

7. Analice el contenido del archivo proc_07.py y ejecutelo. ¿Qué diferencias observa respecto a la creación de procesos en C?

Ejecute con python3

\$ python3 proc_07.py

8. Analice el contenido del archivo proc_08.py y ejecutelo. ¿Qué tipo de variable es la variable suma?

Ejecute con python3

\$ python3 proc_08.py

9. Analice el contenido de los archivo p09a.py, p09b.py, p09c.py y p09d.py. Ejecutelos. Osberve la salida de la consola. ¿Qué diferencias observa con respecto a la creación de archivos en C y con respecto a los archivos de los ejercicios 8 y 9?

Ejecute con python3	<pre>\$ python3 p09a.py</pre>
Ejecute con python3	<pre>\$ python3 p09b.py</pre>
Ejecute con python3	<pre>\$ python3 p09c.py</pre>
Ejecute con python3	<pre>\$ python3 p09d.py</pre>

Procesos huérfanos y zombies en C

10. En el programa del prc_10.c el proceso hijo hace una espera activa de 30 segundos con la función sleep(). El proceso padre termina antes que el proceso hijo.

Compile y ejecute el programa proc_10.c. Ejecute en otra consola:

```
$ pstree -p > pstree.txt
$ ps aux | grep proc 10
```

¿Qué observa a la salida de este comando? ¿Qué sucede con el proceso hijo?.

11. Tome el programa proc_10.c y fuerce a que el proceso hijo haga una espera activa de 30 segundos con la función sleep(). El proceso padre debe terminar antes que el proceso hijo. Compile y ejecute el programa proc_10.c. Ejecute en otra consola:

```
$ pstree -p > pstree.txt
$ ps aux | grep proc 10
```

¿Qué sucede con el proceso hijo?. Identifique si persisten los procesos en cuestión. Observe los números de pid.

12. Tome el programa proc_10.c y fuerce a que el proceso hijo haga una espera activa de 30 segundos con la función sleep() y agregue al final del código del proceso padre la función wait(NULL).

```
$ pstree -p > pstree.txt
$ ps aux | grep proc 10
```

Identifique si persisten los procesos en cuestión. Observe los números de pid.

13. Tome el programa proc_10.c y fuerce a que el proceso padre entre en un bucle infinito con la función while(1). Visualice luego los procesos en ejecución con \$

```
$ pstree -p > pstree.txt
$ ps aux | grep proc 10
```

Identifique el proceso "zombie".

Función execl() en C

- 14. Compile y ejecute prc14.c. ¿Qué sucede al ejecutar la función execl()?
- 15. En el archivo prc14.c, comente la línea 13 y descomente la línea 14. Compile y ejecute. ¿Qué observa por consola? ¿Por qué ha cambiado la salida respecto al anterior?.