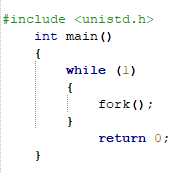
***Manejo de procesos en Linux***

1. ¿Que sucede si en un sistema Linux ejecutamos el siguiente código "fork bomb"?



¿Qué efecto causa el código anterior?

Es un bucle infinito en el cual se empiezan a crear procesos en cada iteración, y cada nuevo procesos creara otro proceso, creando un montón de procesos de forma exponencial, es muy peligroso ya que cada proceso creado consume recursos y memoria, y podría dañar la computadora

¿Linux limita la cantidad de procesos en modo usuario?, ¿Y en modo Kernel?

* Investiga sobre *ulimit –u, pid\_max,* ¿Qué determinan esos parámetros?, ¿Son con figurables?.
* Investiga sobre (OOM Killer), ¿consideras que se activará en este caso?, ¿Porque?

Si, Linux impone limites en el modo usuario y el modo kernel, y lo hace mediante parámetros y mecanismos internos del sistema operativo.

Un usuario norma no puede crear una cantidad ilimitada de procesos ya que esto puede afectar a la estabilidad del sistema

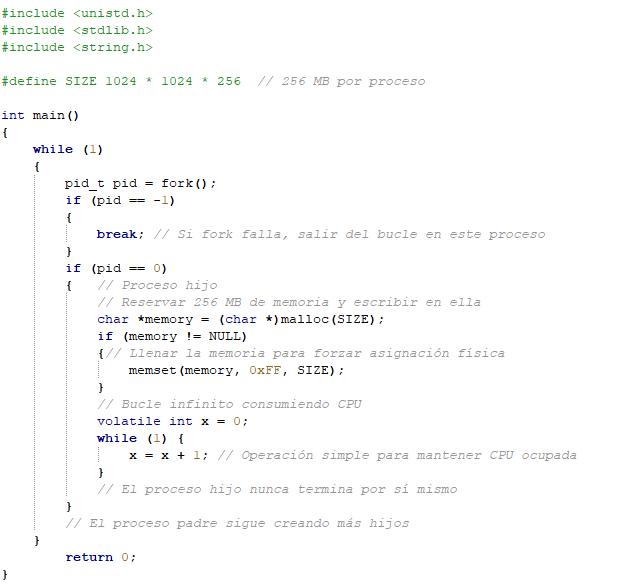
ulimit -u: es un comando que determina la cantidad máxima de procesos(incluyendo hilos) que puede crear un usuario individual, si es con figurable mediante la terminal.

Pid max: el kernel le asigna un pid a cada proceso, y ese numero también tiene un máximo. Este comando limita el pid máximo. que puede asignar el sistema, y limita la cantidad total de procesos(de todos los usuarios). Esto también es configurable a través de la terminal.

OOM Killer: es un mecanismo del kernel de Linux que se activa cuando el sistema se queda sin memoria RAM. El sistema elige y mata procesos para evitar que el sistema deje de funcionar, en este caso es probable que se alcance el pid max o el ulimit, peor también al consumir tanta memoria RAM es probable que OOM KILLER se dispare.

**NOTA:** *Ejecutar el siguiente código en un sistema real puede dañarlo o corromper datos, si el sistema no logra recuperarse. Si lo pruebas, hazlo en un entorno virtualizado (como una máquina virtual) para evitar consecuencias graves.*

1. ¿Qué sucede si en un sistema Linux ejecutamos el siguiente código?



* Si tenemos un sistema con 16GB de RAM y una partición SWAP de 16GB, ¿cuantos procesos puede soportar?. ¿Entra en juego el SWAP? ¿Cuando?
* ¿Que pasa con la unidad de Disco?
* ¿Que ocurre en este caso con (OOM Killer)?

El siguiente código en C implementa un ataque al sistema mediante la creación masiva de procesos que consumen tanto memoria RAM, SWAP como CPU, provocando una situación crítica en el sistema operativo Linux

Cada proceso consume aproximadamente 256 MB de memoria real.

La suma total disponible entre RAM y SWAP es:

16 GB RAM + 16 GB SWAP = 32 GB disponibles

Dividiendo:

32 GB / 256 MB = 128 procesos aproximadamente

El uso intensivo de SWAP genera una actividad de escritura y lectura constante en el disco. Esto provoca ralentización extrema del sistema.

Si se llega al 100 % de uso de SWAP, el disco se vuelve el cuello de botella. También puede afectar otras funciones del sistema que dependan de disco.

El OOM Killer se activa, pero puede no ser suficiente para detener el problema si no se mata también al proceso padre.