

Taller 9

Procesos/Señales

Introducción

En este taller usaremos las llamadas al sistema relacionados a procesos, para acelerar el rendimiento de una aplicación.

Descripción

Este taller se basará en el taller 9, el programa de descarga. Para este taller, haremos uso de la llamada `fork()` para poder atender las solicitudes de manera independiente. Su tarea consiste en implementar esta funcionalidad. Al programa modificado lo llamarán **servidor_multiproceso**. Deben crear un archivo `.c` llamando **servidor_multiproceso.c** donde harán esta implementación.

Para probar su implementación, se les proporciona un script de Python, que generará 50 archivos de entre 1 MB y 10MB. Los archivos tendrán el nombre:

archivo0

archivo1

...

Luego, escribirán un **bash script**, con un lazo de **10000 iteraciones**. En cada iteración, ejecutarán un instancia del cliente:

```
./cliente 127.0.0.1 puerto archivos_aleatorios/archivo# archivo#
```

Donde numeral es un número de 0 a 49, que se incrementa de uno en uno. Una vez que lleguen a 49, vuelven a 0 en la siguiente iteración y así sucesivamente. Corran este script primero cuando **servidor** (del taller 9) esté corriendo.

De manera adicional, deben implementar el manejo de la señal `SIGTSTP` (`Ctrl+Z`) en el servidor. De esta forma, cuando el servidor reciba dicha señal, cerrará el puerto por el cual está escuchando a las conexiones y saldrá. Noten que los hijos que están procesando información deberán bloquear dicha señal (solo el proceso principal manejará esta señal).

Su taller debe incluir un `Makefile`.

Entregable

Repositorio de git del taller, incluido `makefile`. Tarea grupal.