



Universidad Simón Bolívar
Departamento de Computación y
Tecnología de la Información
CI-3825: Sistemas de Operación I
Enero – Marzo 2018

Prof. Fernando Torre Mora
Prof. Carlos Gómez
Prof. Fernando Lovera

Laboratorio de Semana 5

File Merge

Una compañía manejaba su lista de clientes guardando los nombres en un archivo y los apellidos en otro, ya que nunca los usaba juntos. Sin embargo, un cambio en la ley les obliga a usar siempre el nombre completo por lo que deben concatenar ambas listas. La compañía usa en su sistema operativo una política de acceso a disco personalizada que permite cobrarle a sus clientes por el tiempo de uso, por lo que cada microsegundo que el programa de concatenación esté trabajando con los archivos, es un microsegundo que no puede ser cobrado a un cliente.

Usted ha sido elegido para determinar la forma más eficiente de escribir este programa. La compañía ha propuesto los siguientes 3 mecanismos:

1. Hay un proceso padre que crea un archivo de salida y un archivo temporal para comunicarse con sus procesos hijo.
 - a. Se crea un proceso hijo para leer 1 línea de uno de los archivos (dos hijos en total), la cual se escribe en el archivo temporal.
 - b. Cuando el hijo termina, el padre – que se entera que su hijo terminó usando `wait(...)` – crea a un proceso para leer la siguiente línea y, si tiene suficiente en el archivo temporal para armar uno o más nombres completos, los escribe en el archivo de salida.
2. Un proceso padre crea un archivo de salida y un archivo temporal para comunicarse con sus procesos hijo.
 - a. Se crea un proceso por cada archivo, el cual notifica a su padre que escribió en el archivo temporal usando una señal.
 - b. El padre revisa, cada vez que recibe una señal, si tiene suficiente en el archivo temporal para armar uno o más nombres completos, los escribe en el archivo de salida.
3. Un proceso padre crea un archivo de salida, un proceso hijo, y dos *pipes FIFO*.
 - a. Cada hijo escribe en su *pipe* la línea que lee y envía una señal al padre para que revise el *pipe*.
 - b. El padre revisa el *pipe* respectivo al recibir la señal respectiva y, si con eso tiene suficiente para armar un nombre, lo escribe en el archivo de salida.

Puede medir el tiempo de ejecución del programa usando el comando `time` de UNIX para determinar el mejor.

1 Requisitos del programa

Cada programa debe recibir tres parámetros: el archivo de nombres, el archivo de apellidos, y el archivo de salida. El archivo temporal puede ser creado con la función `tmpfile()` de `stdio.h`.

2 Extra Credit

Puede presentar el siguiente programa al final del curso si le faltan puntos:

- Investigue los comandos equivalentes del API de Windows e implemente el equivalente a cualquiera de los tres métodos que puedan correr en el *Command Prompt* (Símbolo de Sistema) de Windows
- Para puntos adicionales, implemente la aplicación de tal manera que pueda ser compilada en Windows o en Linux usando compilación condicional (`#if defined(WINDOWS)`, etc)