Alumno: Roberto Fernández Montero
DNI: 46854179M
Centro Asociado: Móstoles-MadridSur
Email: rfernande1367@alumno.uned.es

Trabajo de Diseño y Administración de Sistemas Operativos

Segunda PED

Introducción

El presente documento tiene como finalidad de actuar como memoria del desarrollo de la segunda práctica de evaluación continua de la asignatura de *Diseño y administración de sistemas operativos.*

Esta primera práctica consiste en un paso de mensajes recibidos a través de teclado utilizando 3 de los mecanismos que linux pone a nuestra disposición, como son las tuberías anónimas, los Fifo (O tuberías con nombre), la cola de mensajes y la memoria compartida.

Dicha práctica debe estar realizada en C para los fichero fuente 1, fuente 2 y fuente3, los cuales explicaré en siguientes apartados, y en Bash para el fichero Ejercicio2 qué sera un script muy sencillo para realizar la compilación y la ejecución del binario.

Para realizar esta práctica el equipo docente nos ha provisto de una máquina virtual con basada en Linux con distribución Ubuntu.
Yo he utilizado Sublime Text para la realización del los ficheros fuente, utilizando el plugin rsync para sincronizar ficheros con mi máquina local y la máquina virtualizada y así poder realizar un desarrollo más rápido.

Alumno: Roberto Fernández Montero
DNI: 46854179M
Centro Asociado: Móstoles-MadridSur
Email: rfernande1367@alumno.uned.es

Implementación

Primeramente se detalla cómo se ejecuta la práctica:

Simplemente será necesario descomprimir el tar enviado como resultado de la práctica y ejecutar el comando ./Ejercicio2.sh desde el directorio raiz del .tar descomprimido.
Este fichero primeramente borrara ejecutables de anteriores ejecuciones si los hubiera. Seguidamente hará lo mismo con el fichero1 si este existiera y por último compilara los ficheros de código fuente, asignándole los permisos necesarios para la ejecución y por último lanzará Ej1 para arrancar el proyecto.

Dicho Ej1 pedirá un mensaje, y este será el mensaje que se enviará desde un proceso a otro, mostrando siempre un mensaje por consola en el cual indica qué fichero se está ejecutando, con qué pid de proceso y dónde está escribiendo el mensaje introducido.

Para la implementación se han utilizado, como se ha comentado antes, librerías estándar del sistema operativo linux, ya que la ejecución del ejercicio tenía que realizarse en una máquina virtual con Ubuntu configurada por el equipo docente, por lo que no ha sido necesario hacer diferentes implementaciones para soportar otros sistemas operativos o implementación de librerías que suplan la inexistencia de otras en dichos sistemas.

Dado qué en el enunciado de la práctica detalla de forma muy concisa lo qué debe hacer el programa, únicamente me he limitado a seguir dicho guión, utilizando los recursos qué linux pone a nuestra disposición para todos los casos, tales como constantes incluidas en ficheros de cabecera o funciones, como por ejemplo los necesarios para el tratamiento de ficheros, errores, memoria compartidas, colas etc.

La gestión de errores se ha realizado de forma muy sencilla ya qué no se especificaba cómo deben tratarse, por lo tanto, en caso de detectar un error (Siempre basándonos en los resultados de las llamadas a sistema y consultando la documentación en la página de referencia man) simplemente se imprime el mensaje utilizando **strerror**.

Para mostrar las estadísticas de ejecución he copiado el código de ejemplo qué aparece en el libro de la asignatura.

Alumno: Roberto Fernández Montero
DNI: 46854179M
Centro Asociado: Móstoles-MadridSur
Email: rfernande1367@alumno.uned.es

Ejecución de ejemplo

Esta es la traza qué genera una ejecución del programa:

```
sh Ejercicio2.sh
Introduce un mensaje: test
El proceso P1 (PID=24043, Ej1) transmite un mensaje al proceso P2 por una tubería anónima
El proceso P2 (PID=24044, Ej1) transmite un mensaje al proceso P2 por una tubería FIFO
El proceso P2 (PID=24044, Ej2) transmite un mensaje al proceso P3 a través de una variable
en memoria compartida
Leída variable de memoria compartida ->  test
El proceso P3 (PID=24045, Ej3) transmite un mensaje al proceso P1 por una cola de mensajes

Tiempo real = 2.65 segundos
Tiempo de uso de la CPU en modo usuario = 0 segundos
```

En esta traza se puede comprobar como, primeramente se pide un mensaje por teclado, ese mensaje se envía a otro proceso mediante una tubería anónima, dicho proceso lo transmite mediante una tubería FIFO después lo comparte en una región de memoria compartida, por ultimo, el ultimo proceso lee dicha variable de la memoria y lo devuelve al primer proceso por una cola de mensajes.

Finalmente se muestran las estadísticas de ejecución .

Bibliografía.

1. https://linux.die.net/man/
2. http://stackoverflow.com/questions/tagged/c