

**Prácticas de
Arquitectura de Computadores**

4º curso de Ingeniería de Telecomunicación

Curso 1999-2000

Septiembre

Introducción

La asignatura Arquitectura de Computadores de 4º curso de Ingeniería de Telecomunicación dispone de 3 créditos asignados a prácticas. Estas prácticas van a constar de 1 ejercicio de programación sobre el sistema operativo Linux. Para la realización de la práctica se dispondrá de los Linux del Laboratorio de Telemática.

Cada grupo de prácticas dispondrá de una cuenta NIS en las máquinas Linux, cuyo nombre será arqXY donde XY corresponderá al número del grupo y la password inicial será dada por el profesor de prácticas previa demanda. Se debe cambiar la password, mediante el comando yppasswd, la primera vez que se entre en el sistema. Por favor, no revelen la password de su cuenta. Los directorios de cada usuario están creados de forma que ningún otro grupo de prácticas pueda acceder a su contenido sin su permiso, por lo que la responsabilidad de las posibles copias será enteramente suya.

Las prácticas deben entregarse antes del día **31 de Agosto del 2000** a las 20:00. El formato de entrega está especificado en el ejercicio de prácticas.

Las notas de prácticas saldrán publicadas en el tablón de anuncios del Laboratorio de Telemática así como en su página web (<http://www.tlm.unavarra.es>) y el anuncio de su aparición se hará por la lista de correo del Laboratorio de Telemática.

Bibliografía

UNIX For Programmers And Users A Complete Guide, G. Glass, Ed. Prentice Hall, ISBN 0-13-061771-7

Advanced Programming In The UNIX Environment, W. Richard Stevens, Ed. Addison-Wesley, ISBN 0-201-56617-7

Beginning Linux Programming, N.Mathew & R.Stones, Ed.Wrox, ISBN 1-874416-68-0

UNIX Distributed Programming, C. Brown, Ed. Prentice Hall

Práctica de Septiembre

Librería mpopen

1.- Descripción

NOMBRE

mpopen, mpclose - E/S de procesos

SINOPSIS

```
#include <stdio.h>
```

```
FILE **mpopen(const char *orden, consta char *tipo);
```

```
int mpclose(FILE **files);
```

```
int mpopencont(void);
```

DESCRIPCIÓN

La función mpopen inicia un nuevo proceso para ejecutar un comando especificado, permitiendo además al proceso padre comunicarse con la entrada o la salida estándar de este nuevo proceso.

mpopen acepta dos cadenas terminadas en cero como parámetros: la primera corresponde al comando a ejecutar con sus opciones y la segunda debe estar compuesta exclusivamente por los caracteres 'r' y 'w' con el siguiente significado:

"r" Se desea poder leer la salida estándar del comando.

"w" Se desea poder escribir en la entrada estándar del comando.

"rw" o "wr" Ambas funcionalidades simultáneamente.

El valor de retorno de mpopen es un array de FILE*. En este array se encontrarán los FILE* necesarios para acceder a la entrada o a la salida estándar del nuevo proceso tal y como se le ha solicitado a la función. Si sólo se ha pedido acceso a uno de los dos canales (entrada o salida) el array de retorno contendrá un solo FILE*. Si se solicita acceso tanto a la entrada como a la salida el array contendrá dos FILE* manteniendo el mismo orden que la cadena tipo, es decir, si se especificó "rw" entonces el primer elemento del array será un FILE* (o stream) de lectura de la salida estándar del proceso y el segundo de escritura sobre la entrada estándar.

Los FILE* del array devuelto son streams normales de E/S de la librería estándar a todos los efectos salvo que deben cerrarse con la función pclose en vez de con fclose.

Los stream devueltos por popen tienen buffer completo (fully buffered) de forma predeterminada.

El argumento orden se emplea para realizar una llamada al sistema de tipo exec. El programa

ejecutable será la primera palabra de la cadena y las restantes serán sus opciones, de las cuales puede haber un número indeterminado estando separadas por uno o más espacios, pudiendo terminar la cadena en un fin de línea o no.

La función `mpclose` espera a que el proceso asociado termine y devuelve el estado de salida del proceso.

Un mismo proceso puede haber realizado más de un `mpopen` antes de querer cerrarlo con `mpclose`.

Globalmente a la máquina se lleva un contador del número de veces que se ha llamado a la función `mpopen` desde la última vez que se reinició la máquina. Este contador es un entero y se puede acceder a su valor mediante la función `mpopencont` que devuelve dicho entero. Este contador se mantiene consistente aunque se ejecute la función `mpopen` desde varios procesos diferentes simultáneamente.

VALOR DEVUELTO

La función `mpopen` devuelve `NULL` si las llamadas a `fork(2)` o `pipe(2)` fallan, o si no puede reservar memoria.

La función `mpclose` devuelve `-1` si la llamada de espera al proceso devuelve un error o se detecta algún otro error.

VEASE TAMBIÉN

```
fork(2), exec(3), pipe(2), waitpid(2), fdopen(3), fclose(3),
stdio(3), fflush(3), fopen(3)
```

2.- Ficheros y ejemplos

En el directorio `$(HOME)/.ficheros/` se han dejado dos ficheros: `mpopen.h` y `ejemplo.c`. El primero es el fichero de cabecera que se debe incluir para compilar un programa que emplee estas funciones sin error. El segundo es un programa de ejemplo que se puede compilar y *linkar* con la librería que contenga a las funciones mencionadas. El programa obtenido a partir de `ejemplo.c` deberá a funcionar sin hacer ningún cambio en el código fuente del mismo.

En el directorio `$(HOME)/solucion/prasept` debe encontrarse un `Makefile` as como todos los ficheros `.c` y `.h` necesarios para crear `mpopen.a`. La acción por defecto del `Makefile` (la cual debe funcionar con solo hacer `make` en ese directorio) debe ser crear `mpopen.a`.

Para la corrección de la práctica se borrarán todos los ejecutables y se recompilará mediante el `Makefile`.

Nota importante

Para la realización de esta librería no se permite el uso de las funciones `popen(3)` y `system(3)`.