SISTEMAS OPERATIVOS II

Tercer curso Ingeniería Técnica de Sistemas. Curso 2004-2005

Práctica 4: Sistema de ficheros en UNIX. Memoria compartida, redirección.

Añadir al shell de las prácticas anteriores las siguientes funciones

oculta cl f1 f2... Guarda los ficheros *f1 f2...* en una zona de memoria compartida de clave cl. Elimina los ficheros del disco. Si la zona de memoria ya existe debe dar un error

info cl Muestra información de los ficheros almacenados en la región de memoria compartida especificada por cl. Para cada fichero la información será: propietario del fichero, grupo del fichero, nombre del fichero, tamaño del fichero permisos y fechas del fichero.

recupera [-d] cl Recupera los ficheros ocultos en la zona de memoria compartida. Con -d elimina la zona de memoria compartida.

elimina cl Elimina la zona de memoria compartida especificada por clave.

com < fich Ejecuta el comando *com* (ejecutable con parámetros) con la entrada estándar redireccionada al fichero *fich*.

com > fich Ejecuta el comando *com* (ejecutable con parámetros) con la salida estándar redireccionada al fichero *fich*

com % fich Ejecuta el comando *com* (ejecutable con parámetros) con el error estándar redireccionada al fichero *fich*.

com1 & com2 Ejecuta *com1* (ejecutable con parámetros) redireccionando su salida estándar a la entrada estándar de *com2* (ejecutable con parámetros).

listar [dir] Muestra los ficheros en el directorio *dir*, en caso de no especificarse *d* se entiende que se refiere al directorio actual. Para cada fichero se mostrará, en una línea, el tamaño el propietario, los permisos (rwxrwxrwx), si es directorio (D), fichero (F) dispositivo de bloque (fi), de caracter (C) o enlace simbólico (L), el número de enlaces y el nombre.

Información detallada de las llamadas al sistema y las funciones de la librería debe obternerse con man (shmget, shmat, read, stat,open, dup, opendir, readdir, pipe)

Los comandos de redirección >, <, % crean un proceso para ejecutar *com* y deben ser compatibles entre sí y con la ejecución en segundo plano.

```
#back a.out parm1 parm2 parm3 < fichero1 > fichero2 % fichero3
```

ejecuta en segundo plano *a.out parm1 parm2 parm3* con la entrada, la salida y el error estándar redireccionado.

FORMA DE ENTREGA

Como en las prácticas anteriores

FECHA DE ENTREGA Viernes 28 Enero de 2005