**SISTEMAS OPERATIVOS II**

Tercer curso Ingeniería Técnica Informática Sistemas . Curso 2008-2009

**Práctica 1:** Procesos en UNIX: Señales.

Comenzar la codificación de un intérprete de comandos (shell) en UNIX, que se irá completando en suscesivas prácticas. El shell debe reconocer los comandos que se datallan ancontinuación, teniendo en cuanta que en dicha descripción:

- Los argumentos entre corchetes [] son opcionales.
- Los argumentos separados por `|` indican que debe ir uno u otro, pero no ambos simultaneamente.
- No debe dilapidar memoria (ejemplo: variable que se asigna cada vez que se llama a una función y no se libera).
- Cuando el shell no pueda ejecutar una acción por algún motivo, debe indicarlo con un mensaje como el que se obtiene con `sys_errlist[errno]` o con *perror()* (por ejemplo, si no puede cambiar de directorio debe indicar por qué).
- En ningún caso debe producir un error de ejecución (segmentation, bus error ...), salvo que se diga explicitamente
- Las direcciones de memoria deben mostrase en **hexadecimal**.
- La información que se nuestra en pantalla no debe incluir en ningún caso líneas en blanco.
- El shell leerá de su entrada estándar y escribirá en su salida estándar, de manera que podría ser ejecutado un archivo de comandos invocando al shell con su entrada estándar redireccionada a dicho archivo.

Los comandos que incluirá el intérprete de comandos en esta primera nos permitirán práctica son

**quit** Termina la ejecución del intérprete de comandos.

**fin** Termina la ejecución del intérprete de comandos.

**autores** Indica los nombres y los logins de los autores de la práctica.

**chdir [dir]** Cambia el directorio actual a *dir*. Si no se le suministra argumento informa del directorio actual.

**prompt *prompt*** Cambia el indicador (prompt) del intérprete de comandos.

**getpid** Muestra el pid del proceso y de su proceso padre

**fork** El intérprete de comandos crea un hijo con fork() y se queda (el propio intérprete de comandos) en espera hasta que dicho hijo (el creado con fork) termine.

**stat fich** Nos da información del fichero *fich* (nombre, tamaño, permisos, fecha de ultimo acceso ...) (Una línea solamente)

**delete fich** Elimina el fichero *fich*

**files [-l][dir]** Lista los ficheros del directorio *dir*. Si no se especifica dir se entiende que es el directorio actual. -l indica que han de darse tambien los detalles de los ficheros (los mismos que en el comando stat)

- **sigaction [-v] [-r] [-p] [-t] [-f] [-dN] [-mSEN1] [-mSEN2] ... S1 [S2] ....** Instala un manejador, mediante *sigaction*, para las señales S1, (S2 ...). El manejador incrementará un contador que indica cuantas veces se ha ejecutado para la señal por la que es invocado. Además se admitirán las siguientes opciones

  -mSEN El manejador ha de ejecutarse la señal SEN enmascarada (miempro sa_mask de la estructura *sigaction*

  -dN El manejador ha de quedar en espera N segundos (con la llamada *sleep*). Debe ser la última instrucción dentro del manejador

  -v El manejador debe imprimir en pantalla el nombre dela señal que se ha recibido (la cual está manejando), cuantas veces se ha recibido (el valor de contador) y la dirección de memoria donde está el parámetro que recibe. De no indicarse -v el manejador NO DEBE IMPRIMIR NADA en pantalla.

  -r El manejador reenvía al proceso la señal para la cual es manejador. En caso de reenviarse la señal, debe hacerse despues de imprimir en pantalla (en caso de que se imprima) y antes de quedarse en espera (en caso de que se haya especificado -sN)

  -t El manejador se instala con el flag SA_RESETHAND

  -p El manejador se instala con el flag SA_ONSTACK

  -f El manejador se instal con el flag SA_NODEFER

**sigget [S1] [S2] ...** No da información del estado de cada señal: manejada (con la dirección del menjador, los flags y la máscara asociada), ignorada o acción por defecto, asi comode si está enmascarada o no. Ejemplo

```
#sigget INT HUP SEGV
INT Enmacarada manejador 0x30045a00 SA_RESTART SA_NODEFER. mascara asociada SIGUP SIGUSR1
HUP No enmascarada Accion por defecto
SEGV No emmascarada Ignorada
```

**sigcont [-z] [S1] ...** Nos informa de los contadores de la señales S1 S2 ..., con -z pone a cero dichos contadores. Si no se especifican señales lo hace para todas (tanto mostrar los contadores como ponerlos a cero)

**sigstack [*tam*]** Establece una pila alternativa para la ejecución de las señales de tamaño *tam*. Si no se especifica tamaño nos informa del tamaño de la pila alternativa y de su dirección .

**sigmax [*n* ]** Establece el número máximo de veces que un manejador se reenvía la señal (valor máximo del contador). Si no se especifica *n* nos indica a cuanto está establecido dicho valor. Un valor 0 indica que no ha límite predefinido

**sigmask [-n] [S1] ...** Enmascara (mediante *sigprocmask* las señales S1 S2 .... Si no se especifican señales nos informa de las que están enmascaradas. Con -n las desenmascara.

**sigignore [S1] [S2] ...** Ignora las señales S1 S2 .... Si no se especifican señales nos informa de las que están ignoradas.

**sigdefault [S1] [S2] ...** Pone las señales S1 S2 ... a su acción por defecto. Si no se especifican señales nos informa de las que están a su accion por defecto.

**bucle** Hace que el shell entre en un bucle infinito. Instala un manejador para SIGINT que permite salir del bucle pulsando control-c para seguir ejecutando el shell.

**segmentation** produce un fallo de segmentación en el shell. (No vale enviar SIGSEGV, tiene que ser un fallo de segmentación de verdad).

**fpe** produce una excepción de la unidad en punto flotante en el shell. (No vale enviar SIGFPE, tiene que ser un fallo de segmentacin de verdad).

## NOTAS

Un shell es basicamente un bucle que imprime una petición de orden (*prompt* en la salida estándar (pantalla), lee una orden de la entrada estándar) teclado y procesa dicha orden.

Para imprimir en la salida estándar puede usarse *printf, puts, putchar, ...* o la llamada al sistema *write* con el descriptor 1 (STDOUT_FILENO); para leer de la entrada estándar *gets, fgets scanf, getchar ...* o la llamada al sistema *read* con el descriptor 0 (STDIN_FILENO). El troceado de la cadena de entrada puede resultar sencillo realizandose con *strtok*

**Información detallada de las llamadas al sistema y las funciones de la librería debe obternerse con man, sección 2 para las llamadas al sistema y sección 3 para las funciones de libreria** (strtok, sigaction, sigprocmask, sigaltstack, stat ...). Todo el manejo de señales debe realizarse con las llamadas al sistema de System V R4 (*sigaction, sigprocmask, ...*)

**Salvo que se diga explicitamente (p.e., comandos *segmentation* y *fpe*, en ningún caso la práctica puede producir error en tiempo de ejecución.**

**FORMA DE ENTREGA** Va a ser utilizado el servicio de recogida de prácticas suministrado por el Centro de Cálculo de esta Facultad y parte del proceso de corrección de las prácticas va a ser automtíco (compilación, listado de practicas entregadas etc) por lo cual deben entregarse **exactamente** como se indica a continuación:

- Se colocará el código fuente de la práctica en el directorio asignado para ello antes de la fecha tope de entrega de la práctica.
- Se entregará UN SOLO fichero fuente por práctica, de nombre pN.c (N el número de práctica). Por ejemplo, para esta práctica será p1.c (en minúsculas).
- Los grupos de prácticas son de **2 (DOS)** alumnos. La práctica SOLO DEBE SER ENTREGADA POR UNO DE LOS MIEMBROS DEL GRUPO
- en el código fuente de la práctica debe figurar como comentario el nombre de los autores **exactamente** en el siguiente formato

```
/*
AUTOR:apellido11 apellido12, nombre1:login_en_el_que_se_entrega
AUTOR:apellido21 apellido22, nombre2:login_en_el_que_se_entrega
*/
```

donde:

1. La palabra autor aparece en mayúsculas.
2. Los apellidos y el nombre de los autores están totalmente en minúsculas.
3. apellidoij representa el apellidoj del componente i del grupo de prácticas.
4. No hay espacios antes y despues de los dos puntos.
5. El login que aparece es el del que entrega la práctica (aparece el mismo login en las dos líneas).
6. Los símbolos de comentarios están en líneas distintas.
7. No debe incluirse la letra ñ ni vocales acentuadas en los nombres

FECHA DE ENTREGA VIERNES 7 NOVIEMBRE 2008