SISTEMAS OPERATIVOS II

Tercer curso Ingeniería Informática. Curso 2010-2011

Práctica 1: Procesos en Unix: Señales

Comenzar la codificación de un intérprete de comandos (shell) en UNIX. Nótese que los comandos aquí descritos deben interpretarse de la siguiente manera

- Los argumentos entre corchetes [] son opcionales.
- Los argumentos separados por `|` indican que debe ir uno u otro, pero no ambos simultaneamente.
- El intérprete de comandos debe aceptar y entender la sintaxis aquí propuesta, pero no tiene que forzarla. (por ejemplo, si hay varios argumentos deben aceptarse en el orden especificado, pero puede resultar mas cómodo de programar asumiendo que pueden ir en cualquier orden)

Además deben tenerse en cuenta las siguientes indicaciones

- **En ningún caso debe producir un error de ejecución (segmentation, bus error . . .)**. La práctica que produzca un error en tiempo de ejecución no será puntuada. Excepcionalmente se admitirá un error en tiempo de ejecución en algunos comandos: en estos casos se indicará explicitamente (***)
- No debe dilapidar memoria (ejemplo: variable que se asigna cada vez que se llama a una función y no se libera). **NO SE REFIERE A DECLARAR LOS ARRAYS DE TAMAÑO PEQUEÑO**
- 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).
- Las direcciones de memoria deben mostrase en **hexadecimal**.
- La información que se nuestra en pantalla no debe incluir en ningun caso líneas en blanco (ni líneas de '*' ni de '=',. . .).
- 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.

Comandos a implementar en esta práctica

exit Termina la ejecución del intérprete de comandos.

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.

getpid [-p] Muestra el pid del proceso. Con -p muestra el pid del proceso padre del shell

cd [dir] Cambia el directorio actual del shell a *dir*. Si no se especifica *dir* muestra el directorio actual del shell

fork El shell crea un hijo y se queda en espera (el shell) a que ese hijo termine.

exec prog arg1 . . . Ejecuta, sin crear proceso (es decir REEMPLAZANDO el código del shell) el programa *prog* con sus argumentos. *prog* representa un ejecutable externo y para poder ser encontrado debe especificarse una trayectoria completa hasta él (comenzando por "/", "./" o "../"). Debe usarse la llamada *execv*.

prog arg1 . . . El shell crea un proceso que ejecuta en primer plano el programa *prog* con sus argumentos. *prog* representa un ejecutable externo y para poder ser encontrado debe especificarse una trayectoria completa hasta él (comenzando por "/", "./" o "../"). Debe usarse la llamada *execv*. **NO DEBE ESCRIBIRSE "prog" PARA EJECUTAR ALGO. "prog" REPRESENTA EL NOMBRE DEL EJECUTABLE**

sigaction [-verbose] [-resend] [-resethand] [-onstack] [-nodefer] [-sleepNN] [-maskSEN1] [-maskSEN2] . . . S1 S2. . . Instala un manejador (utilizando la llamada *sigaction*) para las señales S1, S2 . . . con las siguientes caractrísticas

El manejador incrementa un contador (para cada señal) que indica cuantas veces se ha ejecutado.

-maskSEN El manejador ha de ejecutarse con la señal SEN enmascarada (se añade SEN al miembro sa_mask de la estructura *sigaction*)

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

-verbose El manejador debe imprimir en pantalla el nombre de la 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 -verbose el manejador NO DEBE IMPRIMIR NADA en pantalla.

-resend 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 -sleepNN). Se enviará hasta que el contador de señales recibidas sea es establecido por *maxmanejador*. Si *maxmanejador* es 0, se reenvía continuamente

-resethand El manejador es temporal: se instala con el flag SA_RESETHAND

-onstack El manejador se ejecuta en la pila alternativa: se instala con el flag SA_ONSTACK

-nodefer El manejador puede ser interrumpido por la propia señal: se instala con el flag SA_NODEFER

-restart El manejador se instala con el flag SA_RESTART Ejemplo

```
-> sigaction -resend -nodefer -onstack -sleep10 -maskUSR1 -maskHUP INT SEGV
```

Instala, con los flags SA_NODEFER y SA_ONSTACK, un manejador para SIGINT y SIGSEGV. Dicho manejador se ejecuta con SIGUSR1 y SIGHUP enmascaradas (miembro sa_mask). El manejador cuando se ejecute, además de incrementar el contador de veces que se ha recibido la señal, envia la señal al propio proceso y luego se queda en espera 10 segundos.

maxmanejador **[N]** Establece el número máximo de veces que se reenvía una señal con el manejador (que se ha instalado con la opción *-resend*). Si no se especifica N nos informa de cuál es su valor

siginfo **[S1] [S2] ...** Nos da información del estado de las señales SI, S2 ...: manejada (con la dirección del menjador, los flags y la máscara asociada), ignorada o acción por defecto, asi como de si está enmascarada o no. Si no se especifican señales nos muestra la información de todas de todas. Ejemplo

```
#siginfo 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
```

showcont **[S1] [S2] ...** Nos informa de los contadores de los manejadores de las señales S1, S2 .... Si no se especifican señales nos muestra los contadores de todas.

clearcont **[S1] [S2] ...** Pone a cero los contadores de los manejadores de las señales S1, S2 .... Si no se especifican señales pone a cero los contadores de todas.

sigaltstack [***tam***] [***dir***] Establece una pila alternativa de tamaño *tam* para la ejecución de las señales en la dirección de memoria *dir*. Si no se especifica dirección, se obtendrá una asignando mediante *malloc* del tamaño que se le indica. (/***/ es posible que, especificando alguna dirección de memoria concreta como pila alternativa, se pueda producir un fallo de segmentación al recibir una señal cuyo manejador se ejecuta en dicha pila). Si no se especifica *tam* nos muestra la dirección y el tamaño de la pila alternativa

sigprocmask **[-set|-block|-unblock] S1 S2 . . .** Enmascara o desenmascara (mediante *sigprocmask*) las señales S1, S2 . . . . Si no se especifican señales nos informa de las que están enmascaradas.

- **sigprocmask -set [S1] [S2] . . .** Hace que el conjunto de señales emascaradas del proceso sea (mediante *sigprocmask*) S1, S2 . . . .
- **sigprocmask -block [S1] [S2] . . .** Enmascara (mediante *sigprocmask*) las señales S1, S2 . . . .
- **sigprocmask -unblock [S1] [S2] . . .** Desenmascara (mediante *sigprocmask*) las señales S1, S2 . . . .

sigignore **[-dfl] [S1] [S2] . . .** Ignora las señales S1, S2 . . . . Si no se especifican señales nos informa de las que están ignoradas. Si se especifica *-dfl* pone las señales S1, S2 . . . a su acción por defecto y si además no se indican 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). (/***/ evidentemente produce un error en tiempo de ejecución)

Información detallada de las llamadas al sistema y las funciones de la librería debe obternerse con man (sigprocmask, sigaction, sigaltstack, fork, execv, waitpid, stat, . . .)

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 11 MARZO DE 2011