

SISTEMAS OPERATIVOS II

Tercer curso Ingeniería Técnica Informática Sistemas. Curso 2010-2011

Práctica 3: Procesos en Unix: Señales

Continuar 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 explícitamente (***)
- 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, **NO DEBE TERMINAR SU EJECUCIÓN**, simplemente lo indicará con un mensaje como el que se obtiene con `sys_errlist[errno]` o con `perro()` (por ejemplo, si no puede cambiar de directorio debe indicar por qué).
- Las direcciones de memoria deben mostrarse en **hexadecimal**.
- La información que se muestra en pantalla no debe incluir en ningún 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.

El shell ahora permitirá instalar manejadores para señales, así como enmascararlas o ignorarlas. El shell ha de llevar un contador para cada señal en el

que cuenta las veces que se ha ejecutado el manejador para dicha señal. El manejador cuando se ejecute simplemente incrementará en valor de dicho contador. Opcionalmente imprimirá el valor de dicho contador, la señal que maneja y la dirección del parámetro que recibe. Todo el manejo de señales se hará con las llamadas de System V R4: *sigaction* y *sigprocmask*

IMPORTANTE, si la función de lectura de teclado por parte del shell devuelve un error, el shell NO DEBE TERMINAR. es decir, *no debe haber un código* como este (sobra la llamada *exit()*):

```
if (gets(entrada,MAXENTRADA,stdin)==NULL){
    perror ("imposible leer entrada");
    exit(0);
}
```

Comandos a implementar en esta práctica

senal **|-mask|-unmask|-ign|-dfi|-install|-info|-cont|-clear|-stack**

- **senal -mask** [S1] [S2] ... Enmascara (mediante *sigprocmask* las señales S1, S2 Si no se especifican señales nos informa de las que están enmascaradas.
- **senal -unmask** [S1] [S2] ... Desenmascara (mediante *sigprocmask* las señales S1, S2 Si no se especifican señales nos informa de las que están enmascaradas.
- **senal -ign** [S1] [S2] ... Ignora las señales S1, S2 Si no se especifican señales nos informa de las que están ignoradas.
- **senal -dfi** [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 acción por defecto.
- **senal -install** [-v] [-r] [-p] [-s] [-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 con la señal SEN enmascarada (se añade SEN al miembro *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 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 -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 después 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: no es permanente
- s El manejador se instala con el flag SA_RESTART
- p El manejador se instala con el flag SA_ONSTACK: se ejecuta en la pila alternativa
- f El manejador se instala con el flag SA_NODEFER: puede ser interrumpido por la propia señal

Ejemplo

```
-> senal -install -r -f -p -d10 -mUSR1 -mHUP 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, envía la señal al propio proceso y luego se queda en espera 10 segundos.

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

```
#senal -info INT HUP SEGV
INT Enmascarada manejador 0x30045a00 SA_RESTART SA_NODEFER
      mascara asociada SIGHUP SIGUSR1
HUP No enmascarada Accion por defecto
SEGV No enmascarada Ignorada
```

- **senal -cont [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.
- **senal -clear [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.

- **senal -stack [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 especifican parámetros nos muestra la dirección y el tamaño de la pila alternativa

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 obtenerse con man (sigaction, sigprocmask, sigaltstack ...). Para convertir de nombre de señal a número y viceversa puede utilizarse el código disponible en

<http://www.dc.fi.udc.es/~afyanez/Practicas/sources/senales.c>

FORMA DE ENTREGA Como en prácticas anteriores.

FECHA DE ENTREGA VIERNES 10 DICIEMBRE DE 2010