SISTEMAS OPERATIVOS II. Curso 2005-2006

Práctica 5: Procesos en UNIX. Señales.

Añadir al shell de la práctica anterior las siguientes funciones de manejo de las señales. Utilizar el programa *enviar.c* que se suministra para comprobar el funcionamiento de las señales.

- **manejador [-t] [-v] SEN1 SEN2...** Instala un manejador para las señales SEN1, SEN2 (sin el SIG). Cada vez que se ejecute el manejador incrementará un contador que indica cuantas veces se ha recibido la señal.
  - Los parámetros -t y -v son opcionales.
  - -t el manejador es temporal (se instala con SA_RESETHAND)
  - -v el manejador imprime un mensaje indicando que se ha recibido la señal. (Si no se especifica -v NO DEBE IMPRIMIR NADA).
  - Ejemplo

    ```
    #manejador -v INT HUP SEGV
    #manejador -t ILL
    #manejador -t -v USR1 USR2
    ```
- **manejador2 [-p] [-f] [-v] SEN1 SEN2...** Instala un manejador para la señales SEN1 SEN2... (sin el SIG). Cada vez que se ejecute, el manejador incrementará un contador que indica cuantas veces se ha recibido la señal y además envía al propio proceso la señal para la cual es manejador
  - Los parámetros -p, -f y -v son opcionales.
  - -p el manejador se ejecuta en la pila alternativa.
  - -f el manejador se instala con SA_NODEFER.
  - -v el manejador imprime un mensaje que indica que ha recibido la señal, la dirección de parámetro que recibe así como cuantas veces la ha recibido (el valor del contador para esa señal).
  - Ejemplo

    ```
    #manejador2 -p -v -f INT HUP SEGV
    #manejador2 -f ILL
    ```

```
#manejador -p -v USR1 USR2
```

- **manejador3 SEN [-v] [-dDELAY] [-mSEN1 -mSEN2...]** Instala un manejador permanente para la señal SEN. Cada vez que se ejecute el manejador incrementará un contador que indica cuantas veces se ha recibido la señal. Además manejador se ejecuta con las señales que se le especifican con -m enmascaradas y dentro de él hay una llamada a *sleep*.
  - los parametros -d, -m y -v son opcionales.
  - -dDELAY. DELAY retardo (en segundos) introducido dentro del manejador con una llamada a *sleep*. Si no se especifica se sobreentiende que es 1.
  - -mSENAL. El manejador se ejecuta con SENAL emascarada. (campo *sa_mask* de *sigaction*.
  - -v. El manejador imprime dos mensajes: el primero de ellos al comanzar su ejecución indicando que se ha recibido la señal y el segundo después del retardo (cuando va a terminar su ejecución). (Si no se especifica -v NO DEBE IMPRIMIR NADA).
  - Ejemplo de manejador para SIGINT que se ejecuta con SIGHUP enmascarada y que dentro del manejador se espera 10 segundos.

    ```
    #manejador3 -v -d10 INT -mHUP
    ```

- **pila [bytes]** Establece una pila alternativa para las señales, de tamaño *bytes* para el manejo de las señales. Si no se especifica *bytes* informa de la pila actual de la señales. (en ambos casos se utiliza *sigaltstack*).
- **ignorar SEN1 SEN2...** Ignora la señales SEN1 SEN2....
- **enmascarar SEN1 SEN2...** Enmascara la señales SEN1 SEN2....
- **desenmascarar SEN1 SEN2...** Desenmascara la señales SEN1 SEN2....
- **reset SEN1 SEN2...** Pone la señales SEN1 SEN2... a su acción por defecto
- **seninfo [SEN1 SEN2...]** Para la señal SEN1, SEN2... indica si está enmascarada, ignorada, si tiene instalado un manejador, la dirección del manejador y los flags asociados a ese manejador. Si no se especifica ninguna señal lo indicará para todas. Ejemplo

```
#senales INT
SIGINT enmascarada manejador SI (0x12450) flags SA_RESETHAND
#senales
SIGHUP no enmascarada manejador NO (SIG_DFL) flags
SIGINT enmascarada manejador SI (0x12450) flags SA_RESETHAND
.......
```

- **cont** [**-c**] [**SEN1 SEN2 ...**] Indica cuantas señales SEN1 SEN2 ... se han recibido. Si no se especifica SEN lo indica de todas. Si se indica -c Pone a 0 el contador de las señal SEN1, SEN2... Si además no se especifica SEN1, SEN2.. los pone todos a 0.
- **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).

Información detallada de las llamadas al sistema y las funciones de la librería debe obternerse con man (sigaction, sigprocmask, sigaltstack...). Dado que en linux no existe str2sig, para obtener el número de señal a partir del nombre de la señal puede usarse la función *senal* del programa enviar.c.

Los manejadores deben instalarse con *sigaction* y la señales enmascararse y desenmascararse con *sigprocmask*. SEN1, SEN2 .. representa el nombre de una señal sin el SIG (ejemplo: INT, SEGV, HUP), es decir tal como lo hacen las funciones *sig2str y str2sig*. El estado de las señales debe obtenerse con *sigaction* y *sigprocmask*, ya que almacenarlo en variables del programa podría hacer que tras una llamada a *fork()* o *exec* no fuese correcto.

Debe tenerse en cuenta que las señales interrumpen las llamadas al sistema (muy notorio en las funciones que leen de teclado) haciendo que devuelvan -1 con errno=EINTR y sin realizar la tarea para la que fueron invocadas. Es necesario por tanto, que en el caso de que una llamada al sistema sea interrumpida por una señal la reiniciemos, bien directamente después de comprobar que ha sido interrumpida o bien indicándole al sistema que lo haga automáticamente (por medio de *sigaction*).

FORMA DE ENTREGA

Como en las prácticas anteriores.

FECHA DE ENTREGA VIERNES 9 JUNIO 2006