SISTEMAS OPERATIVOS II

Práctica 3: Procesos en UNIX. Señales.

1. Realizar un programa, *enviar*, que recibe como argumentos el pid de un proceso, en nombre de una señal (sin el SIG), el número de veces que ha de enviarle dicha señal (0 para indicarle que se la envie continuamente o hasta que el proceso termine) y (opcionalmente) el retardo (en segundos) entre que se la envía una vez y la siguiente. El programa NO DEBE IMPRIMIR NADA en pantalla. Ejemplo

   ```
   %enviar 435 INT 100 1
   ```

   Le envía al proceso con pid 435 100 señales SIGINT dejando pasar un segundo después de enviársela cada vez.
   En caso de que no se especifique el cuarto argumento se entenderá que es 0, y NO SE LLAMARA a *sleep* entre una llamada a *kill* y la siguiente. En el momento en que la llamada para enviar seãles devielve -1 (por ejemplo, el proceso al que se le envía ya no existe) *enviar* termina.

2. Añadir al shell de la práctica anterior las siguientes funciones de manejo de las señales.

instman SEN [-t][-v] Instala un manejador para la señal SEN (sin el SIG). Cada vez que se ejecute el manejador incrementará un contador que indica cuantas veces se ha recibido la señal.

- -t el manejador es temporal (se instala con SA_RESETHAND)
- -v el manejador imprime un mensaje indicando que se ha recibido la seal. (Si no se especifica -v NO DEBE IMPRIMIR NADA).

instman2 SEN [-f][-v] Instala un manejador para la señal SEN (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

- -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).

instman2P SEN [-f][-v] Instala un manejador para la señal SEN (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. El manejador se instala para ser ejecutado 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).

pila [bytes] Establece una pila alternativa, de tamaño *bytes* para el manejo de las señales. Si no se especifica *bytes* informa de la pila actual.

ignorar SEN Ignora la señal SEN.

enmascarar SEN Enmascara la señal SEN.

desenmascarar SEN Desenmascara la señal SEN.

reset SEN Pone la señal SEN 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 SEN 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, str2sig)

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 *strsignal, 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*).

Pueden usarse las utilidades de /usr/proc/bin para comprobar la veracidad de lo reportado por el programa.

FORMA DE ENTREGA

Como en las prácticas anteriores.

FECHA DE ENTREGA VIERNES 14 ENERO 2005