

SISTEMAS OPERATIVOS II. Curso 2005-2006

Práctica 2: Procesos en UNIX. Señales.

Añadir al shell de la práctica anterior las siguientes funciones de manejo 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 seal. (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
```

- **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.
- **maxmanejador** [**veces**] Establece *veces* como el límite de veces que un manejador de los instalados con `manejador2` se ejecuta (limita el numero de veces que se puede reenviar la señal). Si no se especifica *veces* nos indica cual es dicho límite. El valor 0 indica ilimitado.
- **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 la 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 obtenerse con `man` (`sigaction`, `sigprocmask`,

str2sig, sigaltstack...)

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*).

En Solaris 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 13 ENERO 2005