SISTEMAS OPERATIVOS II

Tercer curso Ingeniería Informática. Curso 2003-2004

Práctica 2: Procesos en UNIX. Prioridades

Añadir al shell de la práctica anterior las siguientes funciones de manejo de las prioridades

getpid Indica el pid del proceso.

getppid Indica el pid del proceso padre.

prio pid valor Establece la prioridad del proceso pid a valor. Si no se indica valor informa de la prioridad del proceso pid, y si no se indican ni valor ni pid informa de la prioridad del shell. En caso de no poder establecer la prioridad informará de ello

lim pid valor Establece el límite de la prioridad del proceso pid a valor. Si no se indica valor informa de la prioridad del proceso pid, y si no se indican ni valor ni pid informa de la prioridad del shell. En caso de no poder establecer el límite informará de ello.

nice Devuelve el valor *nice* del proceso.

exprio pri cadena Crea un proceso que ejecuta el programa especificado en cadena (cadena representa un ejecutable junto con sus parámetros y podría contener una lista de variables de entorno, como en *LV comando*) estableciendo su prioridad en el valor especificado por pri. Por jemplo, la siguiente orden ejecutaría el programa a.out dentro de un xterm con prioridad -4 y tendría un entorno formado por las variables TERM y DISPLAY

```
# exprio -4 TERM DISPLAY xterm -e a.out
```

La ejecución será en primer o segundo plano según el valor de *segundoplano*. Se creará una lista de todos los procesos que el intérprete de comandos ejecuta en segundo plano (incluidos aquellos en los que no se ha modificado la prioridad). Dicha lista se manejará con los comandos *procs y clearlist* que se detallan a continuación.

procs[-l] Muestra la lista de los pids de los procesos en segundo plano. Si se invoca con -l produce un listado largo: una **sola línea** para cada proceso. La linea debe indicar: el pid del proceso, el estado (activo, terminado o parado) con el valor devuelto (o en su caso la señal), el instante en que se ha lanzado el proceso y la línea de comando con sus argumentos.

clearlist [-s|-n|a] Elimina de la lista de procesos en segundo plano aquellos que han terminado debido a una señal(-s), que han terminado normalmente (-n) o todos los que han terminado (-a).

recursiva [-f] n Invoca a la función recursiva n veces. La función recursiva recibe como parámetro el número de veces que se tiene que invocar; tiene 3 variables, un array automatico de 100 caracteres, un array estático de 100 caracteres y un puntero a caracter.

```
char auto[100];
static char estatico[100];
char * puntero;
```

Lo que hace esta función es (en este orden)

- asignar memoria (mediante *malloc*) al puntero para 100 caracteres.
- imprimir el valor y la dirección del parámetro que recibe.
- imprimir el valor y la dirección del puntero.
- imprimir la dirección de los dos arrays (el nombre del array como puntero).
- invocarse a si misma con (n-1) como parámetro.

Con la opeción -f libera la memoria asignada al puntero (con *free*) antes de llamarse a si misma recursivamente. Si no se especifica -f debe liberar la memoria asignada al puntero después de haberse llamardo recursivamente.

direcciones Imprime las direcciones de las funciones del programa y de todas las variables globales.

Información detallada de las llamadas al sistema y las funciones de la librería debe obternerse con man

Excepto en el comando *nice*, el acceso a las prioridades debe ser siempre con *priocntl*. Tanto el comando prio como el comando lim, cuando se invocan sin argumentos deben mostrar el valor de la prioridad y el valor del límite.

Para comprobar el funcionamiento de las prioridades puede utilizarse el programa e.c disponible en

```
http://www.dc.fi.udc.es/os/~afyanez/Practicas/sources/e.c
```

ejecutando varias instancias del mismo en sendos *xterm* con distintas prioridades. Ejemplo, si *e* es el ejecutable resultado de la compilación de e.c

```
#exprio -1 xterm -e e 1000
#exprio -30 xterm -e e 1000
#exprio -50 xterm -e e 1000
```

nos permitirá comprobar la influencia de la prioridad en la ejecución de procesos. Para una buena comprobación del funcionamiento de las prioridades, esta práctica DEBE ejecutarse en la máquina local (castro).

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 automt́ico (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 p3.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_del_que_entrega_la_practica
  AUTOR:apellido21 apellido22, nombre2:login_del_que_entrega_la_practica
  */
  ```

  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.

FECHA DE ENTREGA VIERNES 7 MAYO 2004