SISTEMAS OPERATIVOS II

Tercer curso Ingeniería Técnica Informática Sistemas . Curso 2008-2009

Práctica 1: Procesos en UNIX: Creación de procesos y ejecución de programas

Comenzar la codificación de un intérprete de comandos (shell) en UNIX, que se irá completando en suscesivas prácticas. El shell debe reconocer los comandos que se datallan a continuación, teniendo en cuenta que en dicha descripción:

- Los argumentos entre corchetes [] son opcionales.
- Los argumentos separados por `|` indican que debe ir uno u otro, pero no ambos simultaneamente.
- No debe dilapidar memoria (ejemplo: variable que se asigna cada vez que se llama a una función y no se libera).
- Cuando el shell no pueda ejecutar una acción por algún motivo, debe indicarlo con un mensaje como el que se obtiene con `sys_errlist[errno]` o con *perror()* (por ejemplo, si no puede cambiar de directorio debe indicar por qué).
- En ningún caso debe producir un error de ejecución (segmentation, bus error . . .), salvo que se diga explicitamente
- Las direcciones de memoria deben mostrase en **hexadecimal**.
- La información que se nuestra en pantalla no debe incluir en ningun caso líneas en blanco.
- 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.

En está practica, el shell deberá además mantener dos listas:

- *ruta de búsqueda*. Es la lista de directorios donde el shell busca los ejecutables externos (lo que en el *bash* y el *ksh* hace la variable de entorno PATH). La implementación es libre (cada uno como quiera) pero NO DEBE implementarse como una variable de entorno. En comando del shell *ruta* muestra y/o manipula dicha lista

- *lista de procesos en segundo plano.* Cada vez que desde nuestro shell cree un proceso en segundo plano, debe incluirlo en una lista de procesos en segundo plano. El comando *procs* muestra y/o manipula dicha lista. La implementación de la lista es libre

Los comandos que incluirá el intérprete de comandos en esta práctica son

ruta [[+|-]**dir**] Manipula y/o muestra la ruta de bśqueda de nuestro intérprete de comandos. Los directorios de esta lista se especificarán sin el caracter '/' al final (salvo el directorio raíz "/"). Se admiten directorios relativos (p.e. "../bin") en esta lista y NO DEBEN convertirse en absolutos.

ruta *+dir* Añade *dir* a la ruta de búsqueda.

ruta *-dir* Elimina *dir* de la ruta de búsqueda.

ruta *dir* Indica si *dir* está en la ruta de búsqueda.

ruta * Vacía la ruta de búsqueda.

ruta Muestra la ruta de búsqueda.

ejemplos

```
#ruta +/usr/local/bin /*aniade /usr/local/bin a la ruta*/
#ruta +.              /*aniade el directorio actual a la ruta*/
#ruta -/              /*elimina el raiz de la ruta*/
```

importapath Añade los directorios contenidos en la variable de entorno PATH a la ruta de búsqueda del shell. (Error muy común: la primera vez que se ejecuta importapath funciona bien, la segunda vez solo añade un directorio)

buscaruta *file* Busca *file* en la lista de directorios que constituyen la ruta de búsqueda del intérprete de comandos y devuelve la trayectoria completa hasta él (análogo al comando *which* del sistema). Si se le indica una trayectoria absoluta (empezando por ./, por / o por ../) no lo buscará en la lista de directorios y solo indicará si dicho fichero existe o no Ejemplo:

```
#buscaruta xtern
xtern: no encontrado
#buscaruta xterm
/usr/X11/bin/xterm
#buscaruta /usr/bin/xclock
```

```
/usr/bin/xclock
#buscaruta ./aa.out
./aa.out: no encontrado
#buscaruta ./a.out
./a.out
```

xec ***[com [args]]*** El intérprete de comandos ejecuta, sin crear un nuevo proceso (reemplaza su código), el programa especificado en *com* con sus argumentos (especificados por *args*). Para poder ser ejecutado, dicho ejecutable debe residir en uno de los directorios de la ruta de búsqueda del intérprete de comandos o bien especificarse la trayectoria completa hasta él (comenzando por /, ./ o ../). Debe usarse la llamada al sistema *execv* y no *execvp*.

Ejemplo

```
#xec /usr/bin/ls -l /home  /*ejecuta /usr/bin/ls*/
  ....
#xec ./a.out -l /*ejecuta a.out,
                 /*que esta en el directorio actual */
......
#xec ls -l   /* para ejecutar esto, es necesario */
              /*que el directorio  /usr/bin,  */
              /*donde esta el programa ls, se haya incluido */
              /*en la ruta de busqueda */
```

[com [args]] Totalmente análogo al comando xec, salvo que ahora el shell no reemplaza su código, sino que crea el proceso que ejecuta dicho comando. El shell además ha de esperar a que dicho prceso termine (la ejecución es, por tanto, en primer plano). Todo lo dicho sobre los ejecutables en el comado *xec* es aplicable aquí.

Ejemplo

```
#/usr/bin/ls -l /home  /*crea un proceso que ejecuta /usr/bin/ls*/
....
#./a.out -l   /* crea un proceso que ejecuta a.out,*/
......        /* que esta en el directorio actual    */
#ls -l        /* para ejecutar esto, es necesario que el directorio */
              /* /usr/bin, donde esta el programa ls, se haya incluido */
              /* a la ruta de busqueda */
```

.

back ***[com [args]]*** Totalmente análogo al anterior, salvo que ahora el shell NO espera a que el proceso creado termine, por lo que la ejecución es en segundo plano. Además el shell añadirá el proceso a una lista de procesos en segundo plano.
Ejemplo

```
#back /usr/X11/bin/xclock  /*crea un proceso que ejecuta
                           /*/usr/X11/bin/xclock en segundo plano*/
....
#back ./a.out -l   /* crea un proceso que ejecuta a.out,*/
......        /* que esta en el directorio actual, en segundo plano    */
#back xterm -e /bin/ksh /* para ejecutar esto, es necesario que el directorio */
              /* /usr/X11/bin, donde esta el programa xterm, se haya incluido */
              /* a la ruta de busqueda */
```

.

procs [**show**|**del**] [[**-n**] [**-s**] [**-p**] [**-a**] |[**pid1** [**pid2**]**...**]]] Muestra (*show*) o elimina (*del*) procesos de la lista de procesos en segundo plano. Los procesos a listar (o eliminar) pueden especificarse mediante sus pids (*pid1*, *pid2* . . .) o mediante:

- -n todos los que han terminado normalmente
- -s todos los que han terminado debido a una señal
- -p todos los que están parados
- -a todos los activos

Si no se especifica *show* o *del* se entiende que quiere listarse. Si no se especifica conjunto de procesos (bien mediante lista de pids o bien mediante las opciones -a, -s . . .) se entenderá que quiere listarse todos los procesos
ejemplo

```
#procs    /*lista todos los procesos*/
#procs -s -p /*lista los procesos que han terminado debido a una senial wue estan parados*/
#procs show -s -p /*igual que el caso anterior */
#procs del 1511 1520 /*elimina de la lista los procesos de pids 1511 y 1520*/
#procs
#procs show
#procs del   /*estas tres listan todos los procesos en segundo plano
```

En el caso de listar, se indicará, para cada proceso, EN UNA SOLA LINEA

- pid del proceso
- estado, uno de los siguientes
  * ACTIVO
  * TERMINADO POR SEÑAL, se indicará además la señal que lo ha parado
  * TERMINADO NORMALMENTE, se indicará el valor devuelto
  * PARADO, se indicará la señal que lo ha parado
- prioridad actual del proceso
- instante de comienzo del proceso
- línea de comando que se está ejecutando

Información detallada de las llamadas al sistema y las funciones de la librería debe obternerse con man (stat, fork, execve, waitpid ...)

Salvo que se diga explicitamente, en ningún caso la práctica puede producir error en tiempo de ejecución.

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 automtíco (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 pN.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 nom-

bre de los autores **exactamente** en el siguiente formato

```
/*
AUTOR:apellido11 apellido12, nombre1:login_en_el_que_se_entrega
AUTOR:apellido21 apellido22, nombre2:login_en_el_que_se_entrega
*/
```

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.
7. No debe incluirse la letra ñ ni vocales acentuadas en los nombres

FECHA DE ENTREGA VIERNES 28 NOVIEMBRE 2007