

SISTEMAS OPERATIVOS II

Tercer curso Ingeniería Informática. Curso 2003-2004

Práctica 1: Procesos en UNIX. Entorno

Comenzar la codificación de un intérprete de comandos en UNIX, que se irá completando en sucesivas prácticas. Dicho intérprete incluirá de momento los comandos que se citan a continuación. Los argumentos entre corchetes [] son opcionales. Los argumentos separados por | indican que debe ir uno u otro, pero no ambos simultáneamente. No debe dilapidar memoria.

autores Indica los nombres y los logins de los autores de la práctica.

fin Termina la ejecución del intérprete de comandos.

exit Termina la ejecución del intérprete de comandos.

vaciaruta Vacía la ruta de búsqueda del intérprete de comandos. La ruta de búsqueda del intérprete de comandos es el conjunto de directorios donde el shell busca los ejecutables. Su misión es análoga a la variable de entorno PATH en el shell del sistema. NO DEBE implementarse con una variable de entorno.

+ruta dir Añade el directorio *dir* al la ruta de búsqueda del intérprete de comandos. El nombre del directorio debe indicarse sin el carácter / al final (Ejemplo **+ruta /usr/bin** y NO **+ruta /usr/bin/**).

-ruta dir Elimina el directorio *dir* de la ruta de búsqueda del intérprete de comandos.

ruta Muestra la ruta de búsqueda del intérprete de comandos.

importapath Añade los directorios de la variable de entorno PATH a la ruta de búsqueda del intérprete de comandos

entorno [-e] Muestra la lista de todas las variables de entorno accediendo mediante el tercer argumento de main. Para cada variable de entorno muestra la variable (nombre=valor), la dirección de memoria donde se almacena la variable (el valor del puntero, e[i]) y la dirección de memoria donde se almacena el puntero (&e[i]). Después de las variables se muestra el valor (como puntero) del tercer argumento de main y de la variable environ así como las direcciones donde se almacenan. Si se invoca como entorno -e hace exactamente lo mismo pero utiliza la variable *environ* en lugar del tercer argumento de main para acceder al entorno. Ejemplo

```

#entorno
.....
.....
ffbefcb8->env[29]=(ffbeff86) PATH=/usr/bin:./usr/local/bin:
ffbefcbc->env[30]=(ffbeffa5) LC_MONETARY=es
ffbefcc0->env[31]=(ffbeffb4) LC_COLLATE=es
ffbefcc4->env[32]=(ffbeffc2) _INIT_NET_STRATEGY=none
ffbefcc8->env[33]=(ffbeffda) _=./entorno
20a34->environ=ffbefc44 | ffbeffc24->env=ffbefc44 | ffbefc20->argv=ffbefc3c
#

```

set [-e] var=valor Establece el valor de la variable de entorno *var*. Accede mediante el tercer argumento de *main*. En caso de invocarse como `set -e` hace exactamente lo mismo pero utiliza la variable *environ* en lugar del tercer argumento de *main* para acceder a la variable de entorno. Si la variable de entorno que pretende modificarse no existe no hace nada.

putenv var=valor Establece el valor de la variable de entorno *var*. Usa la función de librería *putenv*. Si la variable no existe la crea.

get var Devuelve los valores de la variable *var* obtenidos mediante el tercer argumento de *main*, *environ* y la función de librería *getenv*. Para cada caso imprime además la dirección de memoria donde está la variable, y las direcciones donde se almacenan los punteros. Ejemplo:

```

#get TERM
env:      "TERM=vt100" (ffbeff7b) puntero -> ffbefcd8
environ:  "TERM=vt550" ( 2692a) puntero -> 21fb0
getenv:    "vt550" ( 2692f)
#

```

indicador prompt Cambia el indicador (prompt) del intérprete de comandos.

donde ejecutable Busca *ejecutable* en la lista de directorios que constituyen la ruta de búsqueda del intérprete de comandos y devuelve la trayectoria completa hasta él. Ejemplo

```

#donde xterm
/usr/openwin/bin/xterm
#

```

fork El intérprete de comandos crea un hijo y se queda en espera.

exec com El intérprete de comandos ejecuta, sin crear un nuevo proceso (reemplaza su código), el programa especificado en *com*. *com* representa un ejecutable con sus parámetros. Para poder ser ejecutado, dicho eje-

cutable 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*.

exece LV com El shell ejecuta el programa especificado en *com*, en el entorno especificado por LV. *com* representa un ejecutable con sus parámetros. 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 ..). LV representa una lista de nombres de variables de entorno; los valores de dichas variables de entorno deben obtenerse mediante *environ* o *getvar*. Debe usarse la llamada al sistema *execve*. Ejemplo
exece TERM HZ HOME DISPLAY xterm -e a.out
ejecuta *xterm -e a.out* en un entorno que sólo contiene las variables de entorno TERM, HZ, HOME y DISPLAY

comando El intérprete de comandos ejecuta crea un proceso que ejecuta el programa especificado en *comando*. *comando* representa un ejecutable con sus parámetros. 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*.

LV comando El shell ejecuta el programa especificado en *comando*, en el entorno especificado por LV. *comando* representa un ejecutable con sus parámetros. 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 ..). LV representa una lista de nombres de variables de entorno; los valores de dichas variables de entorno deben obtenerse mediante *environ* o *getvar*. Debe usarse la llamada al sistema *execve*. Ejemplo
TERM HZ HOME DISPLAY xterm -e a.out
crea un proceso que ejecuta *xterm -e a.out* en un entorno que sólo contiene las variables de entorno TERM, HZ, HOME y DISPLAY

segundoplano [on | off] Indica si la creación de procesos por parte del intérprete de comandos para ejecutar programas externos debe ser en primer plano o en segundo plano. Afecta a tanto a *comando* como a *LV comando*. Si no se especifica *on* u *off* nos indicará si la ejecución en segundo plano está habilitada

chdir [dir] Cambia el directorio actual a *dir*. Si no se le suministra argumento

informa del directorio actual.

exit Sale del shell.

Información detallada de las llamadas al sistema y las funciones de la librería debe obtenerse con man (exec, chdir, fork, getenv, putenv, strtok...)

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 automático (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 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 16 ABRIL 2004