

SISTEMAS OPERATIVOS II

Tercer curso Ingeniería Técnica Informática Sistemas. Curso 2010-2011

Práctica 2: Procesos en Unix: Ejecución en segundo plano, prioridades y credenciales

Continuar la codificación de un intérprete de comandos (shell) en UNIX. Nótese que los comandos aquí descritos deben interpretarse de la siguiente manera

- Los argumentos entre corchetes [] son opcionales.
- Los argumentos separados por | indican que debe ir uno u otro, pero no ambos simultaneamente.
- El intérprete de comandos debe aceptar y entender la sintaxis aquí propuesta, pero no tiene que forzarla. (por ejemplo, si hay varios argumentos deben aceptarse en el orden especificado, pero puede resultar mas cómodo de programar asumiendo que pueden ir en cualquier orden)

Además deben tenerse en cuenta las siguientes indicaciones

- **En ningún caso debe producir un error de ejecución (segmentation, bus error ...)**. La práctica que produzca un error en tiempo de ejecución no será puntuada. Excepcionalmente se admitirá un error en tiempo de ejecución en algunos comandos: en estos casos se indicará explícitamente (***)
- No debe dilapidar memoria (ejemplo: variable que se asigna cada vez que se llama a una función y no se libera). **NO SE REFIERE A DECLARAR LOS ARRAYS DE TAMAÑO PEQUEÑO**
- Cuando el shell no pueda ejecutar una acción por algún motivo, **NO DEBE TERMINAR SU EJECUCIÓN**, simplemente lo indicará 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).
- Las direcciones de memoria deben mostrarse en **hexadecimal**.
- La información que se muestra en pantalla no debe incluir en ningun caso líneas en blanco (ni líneas de '*' ni de '=',...).
- 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.

El shell debe llevar una lista de directorios (análoga al PATH del *bash*) donde

busca los ejecutables, de manera que cuando se indica un orden al shell la ejecutará si

- la orden es un comando interno del shell (p.e. *autores*, *getpid* ...)
- la orden representa la trayectoria completa a un ejecutable, comenzando por *"/*, *"/.* o *"/..*
- la orden es el nombre de un ejecutable que está en uno de los directorios de la lista antes citada.

La implementación de dicha lista es libre pero NO DEBE implementarse como una variable de entorno. El comando del shell *mipath* muestra y/o manipula dicha lista. Los directorios, salvo el raíz (*"/*) se especificarán sin la barra al final (*/usr/bin* y no */usr/bin/*)

El shell llevará ahora control (mediante una lista) de los procesos que se lanzan desde él en segundo plano. Para cada proceso ha de poder informarnos de su *pid*, instante de creación, línea de comando que está ejecutando, y estado (activo, parado, terminado normalmente o terminado por señal, y en el caso de los terminados, del valor devuelto, o la señal que lo ha terminado). El comando *procesos* nos mostrará dicha lista

Comandos a implementar en esta práctica

prio [-get|-set] [arg. . .] Muestra o establece la prioridad de un proceso o del propio shell

- **prio -get** [*pid*] Muestra la prioridad del proceso *pid*. Si no se especifica *pid* muestra la prioridad del intérprete de comandos
- **prio -set valor** [*pid*] Establece la prioridad del proceso *pid* a valor. Si no se especifica *pid* muestra la prioridad del intérprete de comandos

uid [-get|-set] [arg. . .] Muestra o establece la credencial del shell. En caso de no poder cambiar la credencial, nos lo indicará con el mensaje de error correspondiente. (Recuérdese que los únicos cambios permitidos son cambiar de efectiva a real o salvada) y para ello es necesario que la efectiva y la salvada sean distintas de la real, lo que se consigue ejecutando un fichero *setuid* de otro usuario)

- **uid -get** Muestra las credenciales de usuario (real y efectiva) del proceso. Muestra tanto el número (*uid*) como el login (nombre) asociado
- **uid -set [-l] cred** Establece (si puede) la credencial efectiva de usuario del proceso. si se especifica *-l cred* representa un login, en otro caso el valor numérico de la credencial de usuario

splano prog arg1 ... El shell crea un proceso que ejecuta **EN SEGUNDO PLANO** el programa *prog* con sus argumentos. *prog* representa un ejecutable externo y para poder ser encontrado puede especificarse una trayectoria completa hasta él (comenzando por `"/`, `"/.` o `"/./`) o residir en uno de los directorios de la lista (*mipath*) del shell. Debe usarse la llamada *execv*. Además añadirá el proceso a la lista de procesos en segundo plano del shell.

- Se modificarán la ejecución en primer plano, segundo plano y ejecución sin crear proceso para que se pueda especificar la prioridad indicándolo mediante el símbolo `@` seguido del valor de la prioridad a continuación de la lista de variables que constituyen en entorno de lo proceso.

[splano | ejecutar] prog arg1 ... * LISTAVARIABLES @pri

ejemplos

```
-> ejecutar ./a.out @10
```

```
-> ejecutar ./a.out * TERM HOME DISPLAY NUEVA=nada USER @12
```

```
-> splano /usr/bin/xterm -e csh @6
```

```
-> splano xterm -e csh * TERM HOME DISPLAY USER @-2
```

procesos [-del] [all|term|sig|stop|act] Muestra (o manipula) la lista de procesos en segundo plano del shell.

- **procesos [all|term|sig|stop|act]** Muestra la lista de procesos en segundo plano. Para cada proceso debe mostrar (en una sola línea) su pid, su prioridad, la línea de comando que ejecuta, el instante de inicio y su estado (activo, terminado normalmente, parado o terminado por señal) indicado, en su caso, el valor devuelto o la señal causante de su terminación o parada. Si no se indican argumentos suficientes (p.e. *procesos* sin argumentos) se listarán todos.

- * **procesos all** Muestra todos los procesos

- * **procesos term** Muestra los procesos terminados normalmente

- * **procesos sig** Muestra los procesos terminados por señal

- * **procesos stop** Muestra los procesos parados

- * **procesos act** Muestra los procesos activos

- **procesos -del** Elimina de la lista de procesos en segundo plano los procesos que se le especifican. (Es un comando de manipulación de la lista de procesos en segundo plano, no tiene que terminar los procesos). Si no se indican argumentos suficientes (p.e. *procesos -del* sin argumentos) se listarán todos.

- * **procs -del all** Vacía la lista

- * **procs -del term** Elimina de la lista los procesos terminados

normalmente

- * **procs -del sig** Elimina de la lista los procesos terminados por señal
- * **procs -del stop** Elimina de la lista los procesos parados
- * **procs -del act** Elimina de la lista los procesos activos

Información detallada de las llamadas al sistema y las funciones de la librería debe obtenerse con man (waitpid, setpriority, getpriority, getuid, setuid, getpwent, getpwuid ...)

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_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 19 NOVIEMBRE DE 2010