

SISTEMAS OPERATIVOS II

Tercer curso Ingeniería Informática. Curso 2010-2011

Práctica 3: Procesos en Unix: Credenciales, prioridades y redirección

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, 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).
- Las direcciones de memoria deben mostrarse en **hexadecimal**.
- La información que se muestra en pantalla no debe incluir en ningún 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.

En esta práctica el shell permitira que se redireccione la entrada, la salida y/p el error estándar de los procesos desde el lanzados. También permitirá que se lancen desde él procesos con la prioridad cambiada. Las redirecciones

de la entrada, salida y error estándar deben ser compatibles entre sí, con el cambio prioridad y con la especificación de un entorno; deben además estar disponibles tanto para la ejecución en primer plano, segundo plano y la ejecución sin crear proceso. Además, si el shell recibe un argumento de línea de comando, entenderá que este es el nombre de un fichero y se ejecutará con su entrada estándar redirigida a ese fichero.

Comandos a implementar en esta práctica

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

- **priority** [pid] Muestra la prioridad del proceso *pid*. Si no se especifica *pid* muestra la prioridad del intérprete de comandos. Nos informa de la prioridad obtenida *getpriority* y también de la política de planificación a la que pertenece el proceso obtenida mediante *sched_getscheduler*
- **priority -set valor** [pid] Establece la prioridad del proceso *pid* a valor. Si no se especifica *pid*, cambia la prioridad del intérprete de comandos
- **priority -sched policy pri** [pid] Pone al proceso *pid* en la política de planificación *policy* (OTHER, BATCH, IDLE, RR o FIFO) con una prioridad *pri*. Si no se especifica *pid* lo hace para el intérprete de comandos

uid [-get|-set] [args] Muestra o establece las credenciales de usuario del proceso

- **uid -get** Muestra las credenciales real y efectiva. Muestra, para cada credencial, **EL VALOR NUMÉRICO DE LA CREDENCIAL Y EL LOGIN ASOCIADO**
- **uid -set id** Establece al credencial efectiva de usuario del proceso a *id*. (*id* es el valor numérico de la credencial)
- **uid -set login id** Establece al credencial efectiva de usuario del proceso a *id*. (en este caso, *id* es el login)
- Modificar la ejecución en primer plano, segundo plano y sin crear proceso, para que se pueda especificar una prioridad. La manera de hacerlo será con el símbolo "@" seguido del valor de la prioridad al final de los argumentos al ejecutable (y antes del "&" si lo hubiere)
- Modificar la ejecución en primer plano, segundo plano y sin crear proceso para que se pueda realizar la redirección de la entrada, salida y/o error estándar. La manera de hacerlo es indicando in:*fichero_entrada*, out:*fichero_salida* y/o err:*fichero_error* después de la lista de variables

del entorno y antes del ejecutable La sintaxis completa de la ejecución de programas quedaría de esta manera

```
[exec] [LISTAVARIABLES] [in:f1] [out:f2] [err:f3] prog [arg1...]  
[@pri] [&]
```

prog1 args ... | prog2 args ... [&] Crea un proceso que ejecuta *prog1* con sus argumentos y redirecciona su salida a la entrada del proceso que ejecuta *prog2*. La ejecución es en primer plano o en segundo plano, si se especifica &

- Modificar el shell de manera que si es invocado con un argumento, este argumento es el nombre de un fichero y el shell se ejecutará con su entrada estándar redirigida a ese fichero. Si dicho fichero no existe el shell terminará con un mensaje de error.

Información detallada de las llamadas al sistema y las funciones de la librería debe obtenerse con man (open, close, sup, pipe, setuid, getpwent ...)

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. apellido_i representa el apellido del componente *i* del grupo de prácticas.
4. No hay espacios antes y después 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 29 ABRIL DE 2011