SISTEMAS OPERATIVOS II

Tercer curso Ingeniería Técnica Informática Sistemas. Curso 2011-2012

Práctica 3: Procesos en Unix: 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 explicitamente (***)
- 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 mostrase en **hexadecimal**.
- La información que se nuestra 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.

En está práctica el shell permitira que se redireccione la entrada, la salida y/p el error estándar de los procesos desde el lanzados. 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

- [**segundoplano**|**ejecutar**] [**LISTAVARIABLES**] **prog** [**arg1...**] [**@pri**] [**entrada:f1**] [**salida:f2**] [**error:f3**] 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
  entrada:*fichero_entrada*, salida:*fichero_salida* y/o error:*fichero_error*
  después los parámetros al ejecutable y (si lo hubiere) del cambio de prioridad. Nótese que puede llevar cualquier conjunto de redirecciones: sólo entrada, sólo salida, sólo error, entrada y salida, entrada y error, salida y error, ...
- [**segundoplano**] **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, según se indique *segundoplano*o no
- 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 obternerse con man (open, close, dup, pipe...)

FORMA DE ENTREGA Como en prácticas anteriores

FECHA DE ENTREGA VIERNES 13 ENERO 2012