

Proyecto Semestral

Objetivo: La pareja de trabajo debe ser capaz de aplicar conocimientos sobre creación, permutación, comunicación y sincronización de procesos para crear un programa en C que interprete instrucciones entregadas por un usuario.

Enunciado:

La Shell o intérprete de comandos es la interfaz de usuario fundamental de los sistemas operativos. Se le pide crear un programa en C que intérprete de comandos básicos y simplificado — `myshellin` — con las siguientes características:

1. El intérprete de comandos debe dar soporte a lo siguiente:
 - a. `set <variable> = <string>`: Debe gestionar variables de entorno propias. El usuario puede crear variables e inicializarlas con un string, si la variable existe se sobrescribe.
 - b. `environ`: Muestra todas las variables de entorno utilizadas.
 - c. `echo <comentario>` o `echo <variable>`: Muestra `<comentario>` por pantalla seguido de una nueva línea o muestra el contenido de una variable existente, si no existe la variable muestra un error apropiado.
 - d. `help`: Muestra cómo se utiliza el intérprete de comandos.
 - e. La ruta donde está situado el binario de su `myshellin` deber ser dinámica, es decir, que debe identificar donde está situado y dar paso a que está ruta pueda cambiar.
 - f. `cd <directorio>`: cambia el directorio actual a `<directorio>`. Si el argumento no ha sido especificado, devuelve el directorio actual. Si el directorio no existe se debe mostrar un mensaje de error apropiado. Este comando debe modificar también la variable de entorno PWD.
2. Debe poder ejecutar todos los programas que proporciona un sistema operativo Linux como `ls`, `pwd`, `mkdir`, `cat`, `rm`, `cp`, `mv`, entre otros. Con sus respectivos argumentos. Si el programa no existe, debe mostrar un mensaje apropiado. También puede invocar programas de terceros. Todo lo anterior se debe realizarse por medio de un fork y posterior ejecución del programa en cuestión.
Los argumentos de los programas pueden contener variables de entorno, por lo que antes de ejecutar, debe identificar que existe variables de entorno, si las hay, debe obtener su contenido y usarlo como argumento.
3. El intérprete de comandos debe soportar redireccionamientos de E/S, sobre `stdin` y/o `stdout`. Por ejemplo, la siguiente línea:

```
miprograma arg1 arg2 < entrada > salida
```

ejecutará el programa `miprograma` con los argumentos `arg1` y `arg2`, el flujo de entrada `stdin` se bifurca desde el fichero `entrada` y el flujo de salida `stdout` se bifurca hasta el fichero `salida`. El fichero `entrada` debe existir. En cuanto al fichero `salida`, si no existe se crea.

Condiciones:

1. El trabajo es en parejas, debe reportarlo en la sección en Moodle.
2. **Cualquier indicio de copia y/o plagio entre parejas y foros de internet, será clasificado con nota mínima.**
3. Desarrolle un informe explicando su código a grandes rasgos y el funcionamiento de su myshellin. Debe especificar detalladamente que técnicas y métodos fueron utilizados y en qué casos lo utilizaron.
4. El código fuente debe estar correctamente organizado y comentado.
5. Debe presentar un código bien dividido, es decir, no toda la funcionalidad de un solo archivo .c. Con esto debe utilizar archivo .h e incluir `makefile` para la compilación.
6. Debe enviar todo lo generado y solicitado (menos binario) en un archivo comprimido .zip a la sección habilitada en Moodle antes del día **jueves 15 de julio hasta las 18:00.**