

1 Objetivo

El objetivo de la tarea es escribir un mini-shell en Linux, con la posibilidad de correr un programa en primer plano, o en segundo plano.

2 Se pide

1. Escribir un mini-shell que espera con un "prompt" que el usuario entre un comando (i.e. El nombre de un programa).

Ejemplos:

```
> ./prog arg1 arg2
> /usr/bin/ls /home
>/usr/bin/who
```

No se pide manejar los caracteres "wildcard" (*, ., *, *, *, ?, etc...).

2. El comando para ejecutar debe contener o no el camino completo hasta el programa (PATH), sin tener que manejar caminos por defecto. Por ejemplo se supone que la ejecución del comando "ls" se hará de la siguiente forma:

```
> /usr/bin/ls
> ls
```
3. El shell debe manejar los siguientes errores:
 - El programa que se quiere ejecutar no existe (el archivo no existe, o no es ejecutable).
 - En el caso de una ejecución en primer plano, el shell debe indicar cuál es el código de salida del programa y si corresponde a un error.
4. Debe permitir la ejecución de un programa (proceso) en primer plano o en segundo plano (con o sin argumentos).

Ejemplo:

```
> ./prog (ejecución de prog en primer plano)
> -/prog& (ejecución de prog en segundo plano)
```

Durante la ejecución en primer plano, el shell debe quedarse bloqueado sin poder entrar otro comando hasta que se termine el programa.

En el caso de una ejecución en segundo plano, el shell debe volver al prompt de inmediato y el usuario deber ser capaz de entrar un nuevo comando sin esperar que se termine el programa anterior.

Se prohíbe el uso de la función `system()` de la biblioteca C

3 Ayuda

Deben usar la siguientes llamadas de sistema para realizar la tarea:

- `fork()`, para crear un proceso
- `execv()`, para ejecutar un programa
- `wait()` o `waitpid()`, para esperar la terminación de un proceso

Se aconseja leer el manual de Linux para cada una de estas llamadas de sistema.

Necesitan también usar algunas funciones de la biblioteca C, tal como `strtok()`.