

# Sistemas Operativos

Ingeniería Técnica de Sistemas 2011

## Práctica #3 (Extensión pequeño shell (**minishell**))

Esta práctica es una continuación de la práctica 2. El shell esta compuesto de diversos comandos builtin, y en caso contrario usará el path para ejecutar el programa que se le indique. Como ejemplo se muestra la implementación de los comandos **builtin echo** y **exit**.

### Ejercicio 1 (Obligatorio)Uso de grupos de procesos

El programa creará un nuevo grupo de procesos para cada comando que se ejecute. Un nuevo grupo de procesos se crea con la función `setpgid()`. Una vez creado un proceso, hay que pasarle el control del terminal (función `tcsetpgrp()`). Para saber qué grupo es el dueño de la terminal en un momento dado se usa la función `tcgetpgrp()`.

### Ejercicio 2 (Obligatorio)Comandos de control de trabajos

El shell tiene que realizar el control de trabajos. Esto consiste en que un proceso puede lanzarse a segundo plano (terminar la línea comando con `&`). Un comando puede pararse usando la tecla **Ctrl-Z**. Hay que implementar los comandos builtin:

- **jobs**: lista todos los comandos ejecutandose en ese shell.
- **fg**: Pasa un comando a primer plano.
- **bg**: Pasa un comando a ejecutarse en segundo plano.

Notese que `waitpid()` tiene varios parametros para controlar su uso.

### Ejercicio 3 (Obligatorio)Implementar pipes

Implementar pipes entre varios procesos. Para ello hay que usar la función `pipe()` que crea un pipe que se puede pasar a los dos procesos.

### Ejercicio 4 (Obligatorio)Implementar el comando kill

Implementar el comando `kill -<signal>pid,job` como builtin. Notese que debe aceptar un pid o un número de trabajo. Se puede utilizar la función `kill`.