

## Trabajo Práctico N2

# myshell

<b>Objetivos</b>	<b>1</b>
<b>Introducción</b>	<b>1</b>
<b>Command line prompt</b>	<b>1</b>
<b>Internal commands</b>	<b>1</b>
<b>Program invocation</b>	<b>2</b>
<b>Batch File</b>	<b>2</b>
<b>I/O redirection</b>	<b>2</b>
<b>Background execution</b>	<b>2</b>
<b>Criterios de Corrección</b>	<b>2</b>
Qué se debe Entregar	3

## Objetivos

Con el desarrollo del siguiente Trabajo Práctico (TP), se busca:

- Utilizar mecanismos de creación de Procesos.
- Utilizar mecanismos de concurrencia e IPC.
- Diseñar un intérprete de línea de comandos al estilo Bourne shell.

## Introducción

Este trabajo práctico consta en la elaboración de un programa en lenguaje C sobre GNU/Linux. El trabajo se divide en soluciones incrementales.

## Command line prompt

myshell debe contar con un prompt que contenga el camino al directorio actual.

## Internal commands

myshell debe soportar los siguientes comandos internos:

- `cd <directorio>` - cambia el directorio actual a <directorio>. Si <directorio> no está presente, reporta el directorio actual. Si el directorio no existe se debe imprimir un error apropiado. Además, este comando debe cambiar la variable de entorno `PWD`.
- `clr` - limpia la pantalla
- `echo <comentario>` - muestra <comentario> en la pantalla seguido por una línea nueva. (multiple espacios/tabs pueden ser reducidos a un espacio).
- `quit` - cierra myshell

## Program invocation

Entradas del usuario que no sean comandos internos deben ser interpretados como la invocación de un programa. La misma tiene que ser realizada mediante **fork** y **exec**. No se puede utilizar

## Batch File

myshell debe ser capaz de tomar sus comandos a ejecutar desde un archivo. Por ejemplo, si la shell es invocada con un argumento **myshell batchfile**. El batchfile contiene un conjunto de comandos de línea para que la shell ejecute.

Cuando se alcance el fin de archivo (EOF), myshell debe cerrarse.

Notar que si myshell se ejecuta sin argumento (`./myshell`), se tiene que mostrar el command prompt y se debe esperar a comandos del usuario vía stdin.

## I/O redirection

Se debe soportar redirección de entrada/salida en stdin y/o stdout. Por ejemplo:

*program arg1 ar2 < inputfile > outputfile*

Ejecuta la el programa 'program' con los arguments arg1, arg2. stdin es reemplazado por inputfile y stdout por outputfile.

La redirección debe funcionar para el comando interno **echo**.

## Background execution

Un ampersand & al final de la línea de comando indica que la shell debe retornar al prompt inmediatamente luego de lanzar al programa a ejecutarse.

## Criterios de Corrección

- Se debe compilar el código con los flags de compilación: `-Wall -Pedantic`
- Dividir el código en módulos de manera juiciosa.
- Estilo de código.

- El código no debe contener errores, ni warnings.
- El código no debe contener errores de cppcheck.

## Qué se debe Entregar

- Informe del desarrollo del proyecto.
- Código (funcionando bajo las especificaciones dadas y bajo cualquier caso de test de parámetros de entrada).
- Makefile