

### 1. Objetivo

Implementar mini shell mediante el uso de llamadas al sistema: fork, exec, wait, exit.

### 2. Ejercicios de Laboratorio

#### 2.1. Mini Shell

Implemente un pequeño mini-shell que reciba comandos ingresados por el usuario y los ejecute. Los comandos definidos serán los siguientes:

- ls <parámetro>: listar los archivos y carpetas ubicadas en so2018 de acuerdo al valor del parámetro utilizado en linux.
- cat <nombre\_archivo>: mostrar el contenido del archivo pasado por parámetro.
- nano <nombre\_archivo>: editar el contenido del archivo pasado por parámetro.
- gcc <nombre\_archivo>: compilar el contenido del archivo pasado por parámetro.
- execute <nombre\_archivo>: ejecutar el archivo compilado pasado por parámetro.
- exit -1: Cerrar shell

Utilice la siguiente guía de pasos para elaborar el programa:

```
main(void) {  
  
    printf(">");  
    /*Leer comando*/  
    while (/*Condicion de termino*/) {  
        if (fork() == 0) {  
            /*Ejecutar comando leído y recibir el estado. Imprimir mensaje de  
error en caso de no poder ejecutarse.*/  
        } else {  
            pid = wait(&estado);  
            printf(">");  
            /*Leer nuevo comando*/  
        }  
    }  
}
```

Pistas:

- Utilice la función strcmp() del paquete string.h para comparar el comando ingresado con algún string específico.
- Para los primeros 4 comandos utilice execlp() y para el quinto utilice execl().

```

#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>
#include <string.h>

int main(void){
    int pid, estado;
    char comando[20];
    char parametro[20];

    printf(">");
    scanf("%s %s", &comando, &parametro);

    while(!(strcmp(comando, "exit") == 0 && strcmp(parametro, "-1") == 0)){
        if(fork() == 0){
            if(strcmp(comando, "execute") == 0)
                estado = execl(parametro, 0);
            else
                estado = execlp(comando, comando, parametro, NULL);

            printf("Error en %s %s => %d \n", comando, parametro, estado);
            _exit(1);
        }else{
            pid = wait(&estado);
            printf(">");
            scanf("%s %s", &comando, &parametro);
        }
    }
    return 0;
}

```