

Practica 03

DOCENTE	CARRERA	CURSO
MSc. Maribel Molina Barriga	Escuela Profesional de Ingeniería de Software	Sistemas Operativos

GRUPO	TEMA	DURACIÓN
6	Compilacion en C y C++ en Linux	5 horas

Integrantes

- José Carlos Machaca Vera
- Jhosep Alonso Mollapaza Morocco
- Patrick Andres Ramirez Santos

Índice

1. Ejercicios propuestos	2
1.1. Ejercicio 1	2
1.2. Ejercicio 2	3
1.3. Ejercicio 3	4
2. Cuestionario	5
3. Indice de Source Code	5

1. Ejercicios propuestos

Se deberá de probar, compilar y ejecutar los siguientes códigos:

1.1. Ejercicio 1

Se crea un archivo Cmake para facilitar la compilacion y ejecucion del codigo, este se presenta a continuacion, y que se puede utilizar con los siguientes comandos desde el directorio con los archivos:

```
$ mkdir cmake-build-debug/  
$ cd cmake-build-debug/  
$ cmake .. # Buscar el archivo CMakeLists.txt en el directorio superior  
$ make # Compilar el proyecto  
$ ./E1 # Ejecutar el proyecto
```

Source Code 1: Contenidos Makefile

```
cmake_minimum_required(VERSION 3.27)  
project(E1)
```

```
set(CMAKE_CXX_STANDARD 14)
```

```
include_directories(.
```

```
add_executable(E1  
    main.cpp  
    LinkedList.cpp  
    LinkedList.h  
    ListNode.cpp  
    ListNode.h)
```

proceso.cpp

Este código crea un proceso hijo que imprime un mensaje en pantalla.

Source Code 2: Contenidos Makefile

```
#include <stdio.h>  
#include <sys/types.h>  
#include <sys/wait.h>  
#include <unistd.h>  
  
int main(void) {  
    pid_t pid;  
    // fork a child process  
    pid = fork();  
    if (pid < 0) { /* error occurred */  
        fprintf(stderr, "Fork Failed");  
        return 1;  
    } else if (pid == 0) { /* child process */  
        execlp("/bin/ls", "ls", NULL);  
    } else { /* parent process */  
        // parent will wait for the child to complete
```

```
    wait(NULL);  
    printf("Child Complete");  
}  
  
return 0;  
}
```

ejemplo.cpp

Este código utiliza polimorfismo para detectar el tipo de un objeto e imprimir una función específica en base a ello:

Source Code 3: Contenidos Makefile

```
#include <functional>  
#include <iostream>  
  
class Laboratorio {  
    int num;  
};  
  
class Practica {  
    int a;  
    Laboratorio lab;  
  
public:  
    operator Laboratorio() { return lab; }  
  
    operator int() { return a; }  
};  
  
void function(int a) { std::cout << "funcion (int) ejecutada"; }  
  
void function(Laboratorio la) {  
    std::cout << "Funcion (Laboratorio) ejecutada";  
}  
  
int main() {  
    Practica p;  
    function((Laboratorio)p);  
    return 0;  
}
```

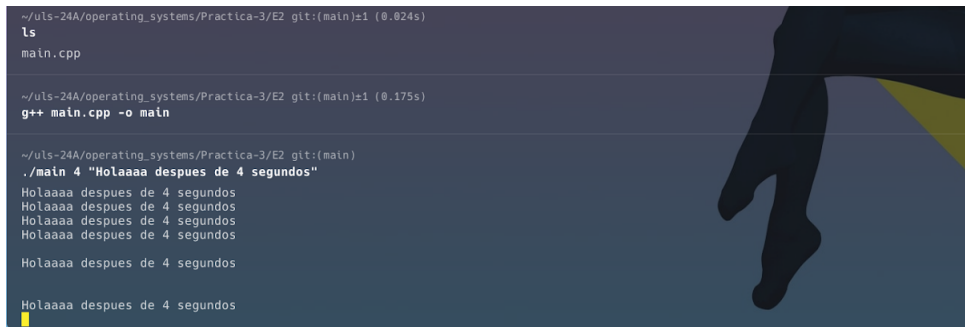
1.2. Ejercicio 2

Este código recibe 2 argumentos vía línea de comandos, el primero es un número de segundos y el segundo es un mensaje, el código espera el tiempo definido por el primer argumento y luego imprime el mensaje en pantalla de forma indefinida.

Source Code 4: E2/main.cpp

```
#include <stdio.h>  
#include <string.h>  
#include <unistd.h>
```

```
int main(int argc, char *argv[])
{
    int segundos;
    if (argc != 3) {
        fprintf(stderr, "Uso: %s <segundos> <mensaje>\n", argv[0]);
        return 1;
    }
    sscanf(argv[1], "%d", &segundos);
    while (1) {
        sleep(segundos);
        printf("%s\n", argv[2]);
    }
    return 0;
}
```



```
~/uls-24A/operating_systems/Practica-3/E2 glt:(main)±1 (0.024s)
ls
main.cpp

~/uls-24A/operating_systems/Practica-3/E2 glt:(main)±1 (0.175s)
g++ main.cpp -o main

~/uls-24A/operating_systems/Practica-3/E2 glt:(main)
./main 4 "Holaaaaa despues de 4 segundos"
Holaaaaa despues de 4 segundos
Holaaaaa despues de 4 segundos
Holaaaaa despues de 4 segundos
Holaaaaa despues de 4 segundos
Holaaaaa despues de 4 segundos
Holaaaaa despues de 4 segundos
```

1.3. Ejercicio 3

En este ejercicio se utiliza un Makefile para compilar el archivo mensaje.c y los archivos que este requiere para ejecutarse, para ejecutarlo se deben seguir los siguientes comandos, luego se muestra el contenido del Makefile:

```
$ make # Compilar el proyecto con el makefile
$ ./mensaje # Ejecutar el proyecto
```

Source Code 5: E3/Makefile

```
mensaje: mensaje.o salida_alt.o
    gcc -o mensaje mensaje.o salida_alt.o

mensaje.o: mensaje.c salida_alt.h
    gcc -c -g mensaje.c

salida_alt.o: salida_alt.c salida_alt.h
    gcc -c -g salida_alt.c
```

```
~/uls-24A/operating_systems/Practica-3/E2 git:(main)±1 (0.024s)
ls
main.cpp

~/uls-24A/operating_systems/Practica-3/E2 git:(main)±1 (0.175s)
g++ main.cpp -o main

~/uls-24A/operating_systems/Practica-3/E2 git:(main)
./main 4 "Holaaaaa despues de 4 segundos"
Holaaaaa despues de 4 segundos
Holaaaaa despues de 4 segundos
Holaaaaa despues de 4 segundos
Holaaaaa despues de 4 segundos
Holaaaaa despues de 4 segundos
Holaaaaa despues de 4 segundos
```

2. Cuestionario

3. Indice de Source Code

Source Code

1.	Contenidos Makefile	2
2.	Contenidos Makefile	2
3.	Contenidos Makefile	3
4.	E2/main.cpp	3
5.	E3/Makefile	4