



Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01 Código: GUIA-PRLE-001 Página: 1

INFORME DE LABORATORIO

INFORMACIÓN BÁSICA						
ASIGNATURA:	Sistema Operativos					
TÍTULO DE LA PRÁCTICA:	Programación en C y C++ Usando el Terminal Linux					
NÚMERO DE PRÁCTICA:	02	AÑO LECTIVO:	2022-В	NRO. SEMESTRE:	VI (sexto)	
FECHA DE PRESENTACIÓN	05/10/2022	HORA DE PRESENTACIÓN	23:59			
INTEGRANTE (s): Yoset Cozco Mauri				NOTA:		
DOCENTE(s): ROL	ANDO JESUS CARD	FΝΔς ΤΔΙ ΔVFRΔ				
DOCERTE(3). NOL	ANDO JESOS CAND	LIVAS TALAVENA				

SOLUCIÓN Y RESULTADOS

I. SOLUCIÓN DE EJERCICIOS/PROBLEMAS

• Creación archivo con gedit

• Compilacion del archivo

```
ncnc@ncnc:~/Desktop/SO$ gcc -o Salida prueba.c
ncnc@ncnc:~/Desktop/SO$ ls
Documentos prueba.c Salida
```

• Ejecución del archivo Salida

ncnc@ncnc:~/Desktop/SO\$./Salida
Ho la mundo desde el Terminal Linux!
ncnc@ncnc:~/Desktop/SO\$

• Compilacion y ejecución de un archivo .CPP





Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01 Código: GUIA-PRLE-001 Página: 2

```
ncnc@ncnc: ~/Desktop/SO

ncnc@ncnc: ~/Desktop/SO$ gcc -o Salida1 prueba.cpp -lstdc++
ncnc@ncnc: ~/Desktop/SO$ ./Salida1
Hola mundo desde el terminal Linux!
ncnc@ncnc: ~/Desktop/SO$
```

• *Compilacion con la herramienta g++*

```
ncnc@ncnc:~/Desktop/SO$ rm Salida1
ncnc@ncnc:~/Desktop/SO$ g++ -o Salida1 prueba.c
ncnc@ncnc:~/Desktop/SO$ ./Salida1
Ho la mundo desde el Terminal Linux!
ncnc@ncnc:~/Desktop/SO$
```

EJERCICIOS PROPUESTOS

• Codigo 1

```
C proceso1.c X
                                                               ₽
lab02 > C proceso1.c > 🕅 main(void)
      #include <sys/types.h>
      #include <stdio.h>
      #include <unistd.h>
      int main(void)
           pid_t pid;
           pid = fork();
           if (pid < 0) { /* error occurred */</pre>
               fprintf(stderr, "Fork Failed") ;
               return 1; }
           else if (pid == 0) { /*x child process */
               execlp("/bin/ls", "ls", NULL);
           else { /* parent process */
               printf("Child Complete");
 22
           return 0;
```

Compilación y Ejecución:

```
ncnc@ncnc:~/Desktop/SO/lab02$ ./proceso1
Child Completencnc@ncnc:~/Desktop/SO/lab02$ proceso1 proceso1.c
```





Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01 Código: GUIA-PRLE-001 Página: 3

```
Codigo 2
 lab02 > @ ejemplo.cpp > ...
        #include <iostream>
        class Laboratorio
            int num;
       };
        class Practica
            int a;
            Laboratorio lab;
            public:
            operator Laboratorio(){ return lab; }
            operator int(){ return a;}
       };
       void funcion (int a)
            std::cout << "funcion (int) ejecutada: ";
       void funcion( Laboratorio lab)
            std::cout << "funcion (Laboratorio) ejecutada: ";
       int main(){
            Practica p;
            funcion(p);
            return 0;
```

Compilacion:

• Codigo 3





Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01 Código: GUIA-PRLE-001 Página: 4





Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01 Código: GUIA-PRLE-001 Página: 5

```
lab02 > 	 LinkedList.cpp > 	 deleteNodes(ListNode *)
      #include "LinkedList.h"
      LinkedList::LinkedList(){
           phead = nullptr;
      void LinkedList::insert(int n){
          if( phead == nullptr){
               phead = new ListNode(n);
               phead-> value = n;
              return;
          else{
               ListNode* pn = new ListNode(n);
              pn-> value = n;
               ListNode* pnode = phead;
              while(pnode-> pnext != nullptr && pnode-> pnext-> value < n){</pre>
                   pnode = pnode-> pnext;
               if(pnode-> pnext != nullptr && pnode-> value > n){
                   pn-> pnext = phead;
                   phead = pn;
               if(pnode-> pnext == nullptr){
                   pnode-> pnext = pn;
              else{
                   pn-> pnext = pnode-> pnext;
                   pnode-> pnext = pn;
      void LinkedList::print(void){
          ListNode* pnodes = phead;
          while(pnodes-> pnext != nullptr){
               std::cout << pnodes-> value << " ";
              pnodes = pnodes-> pnext;
      void LinkedList::deleteAll(void){
          if( phead != nullptr){
               deleteNodes( phead);
      void LinkedList::deleteNodes(ListNode *pn){
          if(pn-> pnext !=nullptr){
              deleteNodes(pn-> pnext);
          delete(pn);
 48
```





Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01 Código: GUIA-PRLE-001 Página: 6

```
lab02 > G ListNode.cpp > ...
                                    lab02 > C ListNode.h > ...
      #include "ListNode.h"
                                          #pragma once
      ListNode::ListNode(){
                                           class ListNode {
          value = -1;
                                               public:
          pnext = nullptr;
                                                   int value;
                                                   ListNode * pnext;
      ListNode::ListNode(int n){
                                                   ListNode();
                                                   ~ListNode();
  9
                                      9
                                           };
#include <iostream>
      #include <stdlib.h>
      #include "LinkedList.h"
      int main(){
          int max = 10;
          LinkedList* plist = new LinkedList
          for(int i = 0; i \le max; i++){
              int num = rand()%max;
              plist->insert(num);
          plist->print();
          plist->deleteAll();
          delete(plist);
          return 0;
 17
```

Compilación archivo main.cpp

```
ncnc@ncnc:~/Desktop/SO/lab02$ touch LinkedList.cpp
ncnc@ncnc:~/Desktop/SO/lab02$ touch ListNode.cpp
ncnc@ncnc:~/Desktop/SO/lab02$ touch main.cpp
ncnc@ncnc:~/Desktop/SO/lab02$ touch ListNode.h
ncnc@ncnc:~/Desktop/SO/lab02$ g++ -o main main.cpp
/usr/bin/ld: /tmp/cc4KbAdM.o: in function `main':
main.cpp:(.text+0x27): undefined reference to `LinkedList::LinkedList()'
/usr/bin/ld: main.cpp:(.text+0x51): undefined reference to `LinkedList::insert(int)'
/usr/bin/ld: main.cpp:(.text+0x69): undefined reference to `LinkedList::print()'
/usr/bin/ld: main.cpp:(.text+0x75): undefined reference to `LinkedList::deleteAll()'
collect2: error: ld returned 1 exit status
ncnc@ncnc:~/Desktop/SO/lab02$
```





Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Aprobación: 2022/03/01 Código: GUIA-PRLE-001 Página: 7

II. SOLUCIÓN DEL CUESTIONARIO

1. ¿Cuál es la diferencia entre compilar con GCC y G++?

GCC es compilador de GNU C mientras que G++ Compilador de GNU C++

Gcc compila archivos *.c *.cpp archivos C y C++. G++ compila archivos *.c *.cpp pero todos los archivos serán tratados como archivos c++. G++ si se usa para vincular archivos de objetos, se vinculan automáticamente en las bibliotecas estándar de c++ (g++ no hace esto)

Gcc compila archivos C pero tiene menos macros predefinidas.

Gcc compila archivos *.cpp y g++ compila *.c *.cpp

2. ¿En qué se diferencia el archivo generado .o contra un .exe?

Un archivo .o (object) es un archivo generado después de compilar código fuente, mientras que un archivo .exe(ejecutable) es un archivo generado después de vincular un conjunto de archivos de objeto mediante un enlazador.

III. CONCLUSIONES

Al hacer referencias en C, todas estas referencias deben existir antes de la compilación, aun así después de probar llenando archivos vacíos, al usar funciones de estas referencias, sigue dando errores, se necesita un poco de conocimientos adicionales en C para poder corregir estos errores.

REFERENCIAS Y BIBLIOGRAFÍA

RETROALIMENTACIÓN GENERAL