



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorios de docencia

Laboratorio de Computación Salas A y B

Profesor: Rodríguez Espino Claudia Ing.

Asignatura: Fundamentos de Programación.

Grupo: 1104

No de Práctica(s): 6

Integrante(s): Santa Rosa Ortiz Thelma Jazmín.

*No. de Equipo de
cómputo empleado:* 50.

Semestre: 1°

Fecha de entrega: 22 de septiembre del 2018

Observaciones: _____

CALIFICACIÓN: _____

Practica 6. Guía práctica de estudio 06: Entorno de C (Editores, Compilación y ejecución)

Objetivo:

Conocer y usar los ambientes y herramientas para el desarrollo y ejecución de programas en Lenguaje C, como editores y compiladores en diversos sistemas operativos.

Desarrollo:

La profesora nos dio a conocer los objetivos y comenzamos a leer la práctica, los conceptos que debíamos saber para llevar a cabo las actividades necesarias para realizar esta práctica.

Un lenguaje de programación permite expresar una serie de instrucciones que podrán ser realizadas por una computadora. Uno de los lenguajes de programación mayormente difundidos es el lenguaje C.

Éste es muy utilizado ya que la forma de dar instrucciones es muy cercana a lo que un humano podría abstraer, es decir, las instrucciones no son tal cual las que una computadora podría entender, para ello se necesitaría conocer a fondo el microprocesador, el sistema operativo entre otros aspectos. Por esta razón, C es conocido como un lenguaje de alto nivel, esto significa a que las instrucciones podrían ser entendidas fácilmente por un humano. En contraparte, un lenguaje de bajo nivel, son instrucciones que son cercanas a lo que la máquina puede entender y difícilmente pueden ser comprendidas por una persona que no tenga conocimientos de la máquina en que operarán. Algunos autores consideran al lenguaje C como un lenguaje de mediano nivel, ya que no es totalmente transparente sino tiene elementos que tienen que ver con la arquitectura de la máquina a la hora de programar.

Un programa en C se elabora describiendo cada una de las instrucciones de acuerdo a las reglas definidas en este lenguaje en un archivo de texto para después ser procesadas en un compilador. Un compilador es un programa que toma como entrada un archivo de texto y tiene como salida un programa ejecutable, éste tiene instrucciones que pueden ser procesadas por el hardware de la computadora en conjunto con el sistema operativo que corre sobre ella. Se tiene como ventaja que un programa escrito en lenguaje C, siguiendo siempre su estándar, puede correr en cualquier máquina siempre y cuando exista un compilador de C hecho para tal.

Editores de C

Un programa en C debe ser escrito en un editor de texto para después generar un programa ejecutable en la computadora por medio de un compilador. Tanto el editor de texto como el compilador van de la mano con el sistema operativo y si posee o no interfaz gráfica por lo que son factores que se deben de tomar en cuenta a la hora de elegir el entorno para desarrollar programas en C.

Es importante señalar que no es lo mismo un editor de texto que un procesador de texto. El primero edita un texto plano que puede tener muchas utilidades como guardar una configuración, tener escrito un programa, etc., y será interpretado hasta que se haga una lectura de éste.

Editor Visual Interface de GNU/Linux (VI)

El editor vi (visual interface) es el editor más común en cualquier distribución de sistemas operativos con núcleo basado en UNIX. Está disponible en línea de comandos y si el sistema operativo tiene entorno gráfico se puede acceder a él desde la terminal.

VI es un editor que puede resultar difícil de usar en un inicio. Aunque existen editores más intuitivos en su uso, en muchas ocasiones VI es el único disponible.

Para iniciar VI, debe taclearse desde la línea de comandos:

`vi nombre_archivo[.ext]`

Después de conocer la parte teórica impórtate la profesora nos guiaba con otros conceptos mientras llevábamos a cabo las actividades a la par de ella.

Modo comando

Es el modo por defecto de VI cuando se abre. Las teclas presionadas ejecutan diversas acciones predeterminadas y no se puede editar el texto libremente. Los comandos son sensitivos a las mayúsculas y a las minúsculas. Algunos ejemplos son:

- ⌘ ↑ o **k** mueve el cursor hacia arriba.
- ⌘ ↓ o **j** mueve el cursor hacia abajo.
- ⌘ ← o **h** mueve el cursor hacia la izquierda.
- ⌘ → o **l** mueve el cursor hacia la derecha.
- ⌘ **1G** lleva el cursor al comienzo de la primera línea.
- ⌘ **G** lleva el cursor al comienzo de la última línea.
- ⌘ **x** borra el carácter marcado por el cursor.
- ⌘ **dd** borra o corta la línea donde está el cursor.
- ⌘ **ndd** donde n es la cantidad de líneas que se borrarán o cortarán después del cursor.
- ⌘ **D** borra o corta desde la posición de cursor hasta el final de la línea.
- ⌘ **dw** borra o corta desde la posición del cursor hasta el final de una palabra.
- ⌘ **yy** copia la línea donde está el cursor.
- ⌘ **p** pega un contenido copiado o borrado.
- ⌘ **u** deshace el último cambio.

Modo insertar

Este modo permite insertar texto. Las teclas presionadas ya no harán una acción como en el modo comando sino será el contenido que formará el texto del documento. Se puede desplazar con las flechas del teclado y borrar con la tecla de retroceso o de suprimir.

Para ingresar al modo insertar existen varios comandos:

- ⌘ **i** pasa al modo insertar poniendo el texto a la izquierda del cursor.
- ⌘ **a** pasa al modo insertar poniendo el texto a la derecha del cursor.
- ⌘ **A** pasa al modo insertar colocando el texto al final de la línea donde el cursor se encuentra.

- + I pasa al modo insertar colocando el texto al principio de la línea donde el cursor se encuentra.
- + O coloca una línea arriba de la línea seleccionada por el cursor y pasa al modo insertar.
- + o coloca una línea debajo de la seleccionada por el cursor y pasa al modo insertar.

Para salir del modo insertar debe presionarse la tecla Esc. Para verificar que se encuentra en modo insertar es se puede ver `-- insertar --` en la última línea del editor.

Compiladores

Una vez codificado un programa en C en algún editor de texto, éste debe ser leído por un programa que produzca un archivo ejecutable. A este programa se le conoce como compilador y depende totalmente del hardware de la computadora y el sistema operativo que corre sobre ella.

Recordando, un programa en C es universal, por lo que cada una de las instrucciones que lo conforman debe poder entenderlas muchos de los equipos en el mercado, aunque su naturaleza sea distinta. Por ello, un compilador depende del equipo, porque es un traductor que transforma ese lenguaje universal a un programa ejecutable que sólo puede correr ese equipo.

GCC (GNU Compiler Collection)

Es un conjunto de compiladores de uso libre para sistemas operativos basados en UNIX. Entre sus compiladores existe el que sirve para programas escritos en C. Se encuentra por defecto en diversas distribuciones de Linux. El compilador trabaja en línea de comandos.

Existe también una versión modificada que puede correr y crear programas para plataformas Windows en un paquete llamado MinGW (Minimalist GNU for Windows).

Al compilar un programa en C el compilador genera diversos archivos intermedios que corresponden a las distintas fases que realiza. Éstas no son de interés por el momento y son eliminadas una vez obtenido el archivo ejecutable. GCC tiene diferentes opciones de ejecución para usuarios más avanzados.

Suponiendo que se tiene un programa escrito en C y se le llamó `calculadora.c` la manera de compilarlo es localizándose mediante la línea de comandos en la ruta donde el archivo se encuentra y ejecutando el comando:

`gcc calculadora.c`

Esto creará un archivo `a.out` (en Windows `a.exe`) que es el programa ejecutable resultado de la compilación.

Si se desea que la salida tenga un nombre en particular, debe definirse por medio del parámetro `-o` de gcc, por ejemplo, para que se llame `calculadora.out` (en Windows `calculadora.exe`):

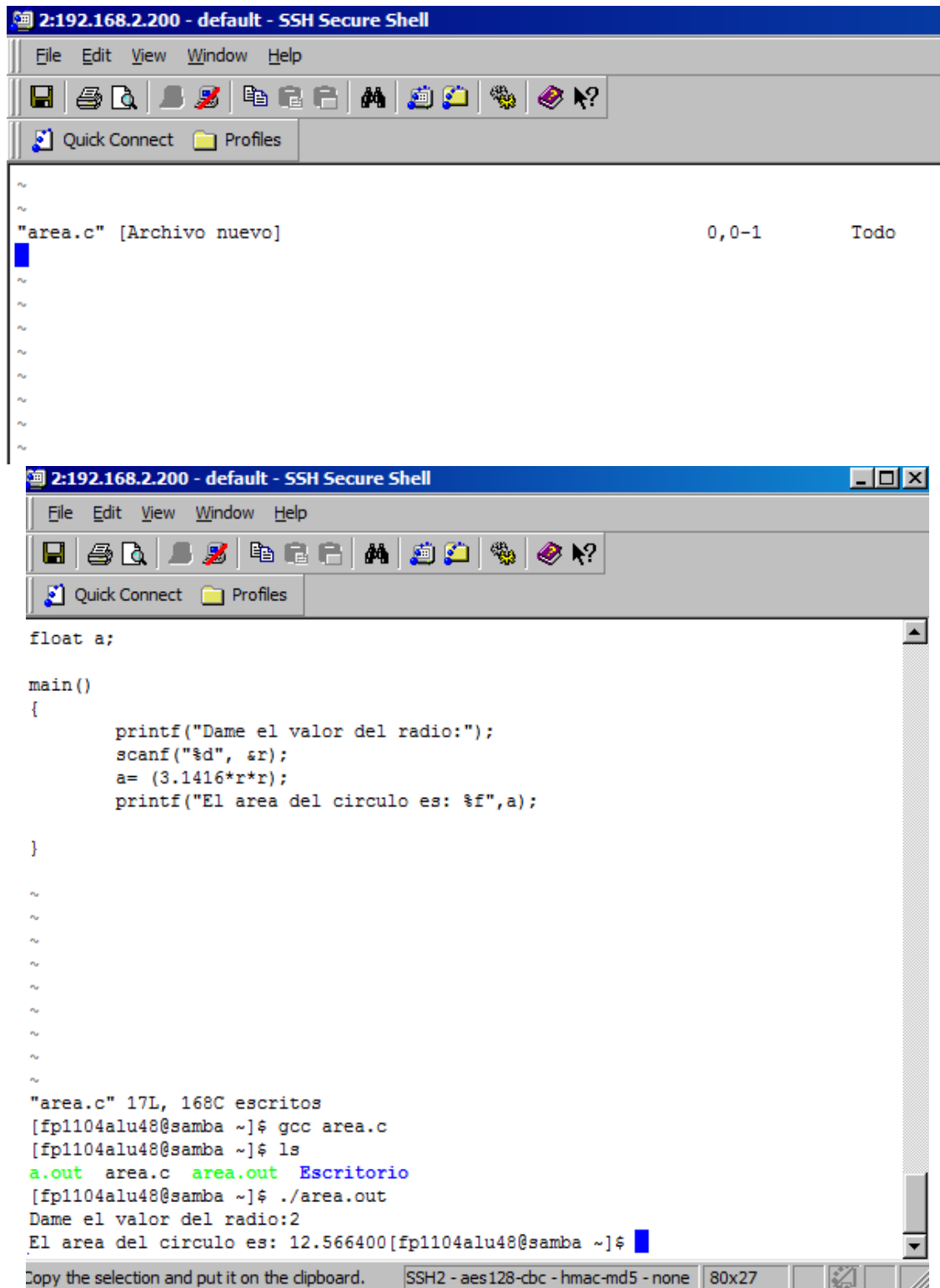
`gcc calculadora.c -o calculadora.out`

A veces, para realizar un programa más complejo, se necesitan bibliotecas que se instalaron en el equipo previamente y se definió su uso en el programa escrito en C pero al momento de compilar es necesario indicar a GCC que se está usando bibliotecas que no se encuentran en su

repertorio de bibliotecas estándar. Para ello es necesario utilizar el parámetro `-l` seguido inmediatamente por el nombre de la biblioteca, sin dejar espacio alguno:

`gcc calculadora.c -o calculadora -lnombre_libreria`

AREA DEL CIRCULO.



```
2:192.168.2.200 - default - SSH Secure Shell
File Edit View Window Help
[Icons]
Quick Connect Profiles

~
~
"area.c" [Archivo nuevo] 0,0-1 Todo
~
~
~
~
~
~
~
~
~
~

2:192.168.2.200 - default - SSH Secure Shell
File Edit View Window Help
[Icons]
Quick Connect Profiles

float a;

main()
{
    printf("Dame el valor del radio:");
    scanf("%d", &r);
    a= (3.1416*r*r);
    printf("El area del circulo es: %f",a);

}

~
~
~
~
~
~
~
~
~
~

"area.c" 17L, 168C escritos
[fp1104alu48@samba ~]$ gcc area.c
[fp1104alu48@samba ~]$ ls
a.out area.c area.out Escritorio
[fp1104alu48@samba ~]$ ./area.out
Dame el valor del radio:2
El area del circulo es: 12.566400[fp1104alu48@samba ~]$

Copy the selection and put it on the clipboard. SSH2 - aes128-cbc - hmac-md5 - none 80x27
```

SUMA DE DOS NUMEROS.

```
suma.c:13:1: error: expected â;â before â}â token
[fp1104alu47@samba ~]$ vi suma.c
#include <stdio.h>

int a,b,sum;

main()
{
    printf("Introduce el primer numero (entero): ");
    scanf("%d",&a);
    printf("Introduce el segundo numero (entero): ");
    scanf("%d",&b);
    sum=(a+b);
    printf("La suma es: %d",sum);
}
~
~
~
~
~
~
~
"suma.c" 13L, 223C escritos
[fp1104alu47@samba ~]$ gcc suma.c
[fp1104alu47@samba ~]$ ls
a.out area.c area.out Escritorio suma.c
[fp1104alu47@samba ~]$ gcc suma.c -o suma.out
[fp1104alu47@samba ~]$ ./suma.c
-bash: ./suma.c: Permiso denegado
[fp1104alu47@samba ~]$ ./suma.out
Introduce el primer numero (entero): 1
Introduce el segundo numero (entero): 1
La suma es: 2[fp1104alu47@samba ~]$
```

ECUACIONES.

```
int x,y;

main()
{
    printf("Dame un valor");
    scanf("%d",&x);

    if(x>2)
    {
        printf("Se resolvera y=x^2+3x-2");
        y=((x*x)+(3*x)-2);
        printf("El resultado es: %d",y);
    }

    else
    if(x<2)
    {
        printf("Se resolvera y=2x^2+x+3");
        y=((2*x*x)+(x+3));
        printf("El resultado es: %d",y);
    }
    else
    {
        printf("No hay solucion cuando el valor es 2");
    }
}

~
~
~
~
~
~
~
~
~
~
~
~

"ecuacion.c" 28L, 366C escritos
[fp1104alu47@samba ~]$ gcc ecuacion.c
[fp1104alu47@samba ~]$
```

Conclusión:

Esta practica me gusto mucho porque ya estamos viendo programación con el lenguaje de C, en ella conocimos y usamos ambientes y herramientas para desarrollar y ejecutar programas (área del círculo, ecuaciones y suma de dos números) de este lenguaje con un editor y un compilador diferente, en este caso fue GNU/Linux.