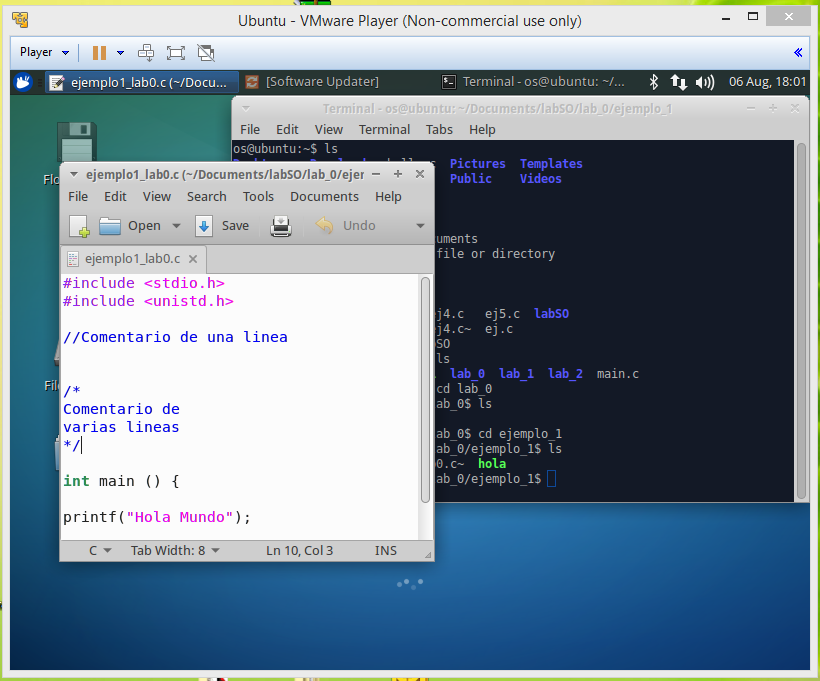
**GUIA BONIFICACION**

1. /dev
2. ../../usr
3. ../../../usr
4. ../../opt/gnome
5. ../share/doc/packages
6. ../../../local
7. ./doc/packages
8. /
9. /home/jono/work
10. /usr/share/doc/packages
11. **Absoluta:** cd /usr/share/doc/packages **Relativa:** cd ./doc/packages
12. **Absoluta:** cd /usr/local **Relativa:** cd ../../usr/local
13. **Desde /usr/lib:** mkdir ../../var/lab
14. mkdir ./a b c
15. mkdir ./photos/1 2
16. rmdir ./photos/1
17. rm-r dir ./c
18. Que en la segunda también se listan los archivos oculos (que empiezan por .)
19. Juntando ambas expresiones con : ls –al
20. Se realizo
21. gcc main.c –o myexe
22. ./myexe
23. sale un error: “myexe: no such file o directory”
24. El programa se escribió y se compilo:



**C**

1. En la programación imperativa **se** **describen los pasos necesarios para solucionar el problema.** Java es un lenguaje imperativo.
2. En la programación imperativa se describe paso a paso un conjunto de instrucciones que deben ejecutarse para variar el estado del programa y hallar la solución. **Se** **describen los pasos necesarios para solucionar el problema**

En la programación declarativa las sentencias que se utilizan lo que hacen es **describir el problema** que se quiere solucionar, pero no las instrucciones necesarias para solucionarlo. **Esto último se realizará mediante mecanismos internos de inferencia de información**

Prolog es un lenguaje de declaración declarativo.

1. C# tiene Garbage collector y funciona para limpiar memoria de información inútil, como objetos carentes de dirección, para agilizar el programa y hacerlo más eficiente.
2. Es una recopilación de ficheros cabecera y bibliotecas con rutinas, funciona para implementar operaciones comunes, tales como las de entrada y salida o el manejo de cadenas
3. Uno compila a código de máquina y el otro a byte-code.

DECLARACION DE LA VARIABLE “Main”

DECLARACION DE LA FUNCION “funcPrint”

FIN DEL PROGRAMA

IMPRESIÓN DE LA CADENA “Hola Mundo Cruel”

LLAMADA A LA FUNCION “funcPrint”, CON EL PARAMETRO “Hola Mundo Cruel”

1. ¿?
2. 01001000 01101111 01101100 01100001 00100000 01001101 01110101 01101110 01100100 01101111 00101111 01110010 00101111 0110111
3. k = 6

k = 5

k = 6 i = 20

k = 6 i = 24

1. nl, /n
2. ¿?
3. El programa le pide al usuario escribir los caracteres que desee y luego los imprime uno en cada línea, además, si se escribe X, el programa para, al final se imprime end of program. 2 nuevas líneas, una dentro del do y otra cuando el programa es finalizado, antes de el “End of Program”.
4. El resultado de la impresión es:

w

w

1

29

29

0000000029

29 w

35

1d

Cada línea al final realiza un salto de línea

La primera línea simplemente imprime lett

La segunda línea imprime lett en la cuarta posición

La tercera línea imprime i

La cuarta línea imprime j

La quinta línea imprime j en la decima posición

La sexta línea imprime ceros hasta la decima posición donde imprime j

La séptima línea imprime j y 10m espacios después imprime lett

La octava línea imprime un entero octal sin signo

La novena línea imprime un entero hexadecimal sin signo

**APUNTADORES**

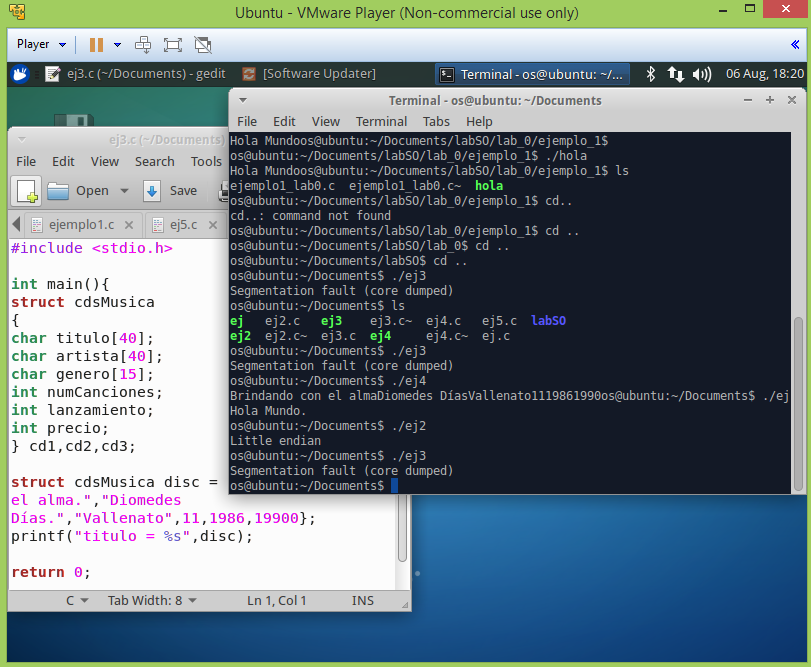
1. pvar2 = &var2;
2. pvar1 = pvar2;
3. px = &x;

y = \*px;

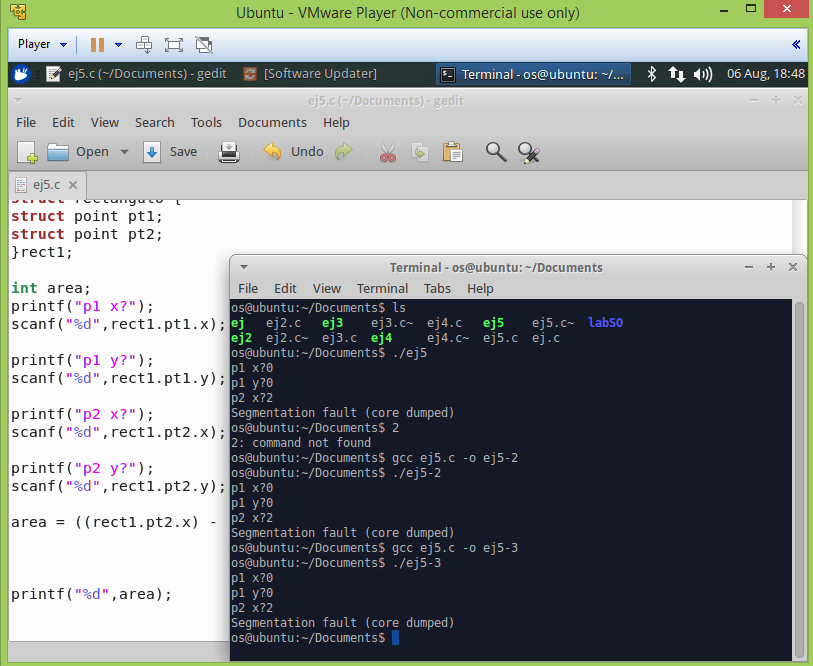
1. px apuntaría a la dirección de la memoria anterior a 4, 0 y le daría un valor de 5 a esa posición/variable.
2. a= 2, b = 4, c = -1, d = -2
3. Espacio de memoria que almacenan datos, como variables el STACK se usa para seguir el estado actual de programa y almacenar variables locales
4. Para pasar los parámetros con la dirección de las variables y no las variables en sí, en la b la respuesta correcta es la a.
5. Los errores que aparecieron estaban relacionados con los punteros y los int, el primero es que la función f1 espera recibir una int y 2 punteros, pero recibe 3 int, el segundo es que se realizan operaciones en las que se comparan los punteros con las int.
6. ¿?
7. ¿?
8. ¿?
9. ¿?
10. Es un operador que proporciona el tamaño en bytes de un tipo de valor o variable
11. El programa primero declara dos punteros de tipo char, a la ves que determina el valor reservado en la dirección de uno de ellos. Luego le asigna a buffer la dirección en memoria reservada por el malloc, del tamaño en bytes de un char por 30. Continua copiando en la dirección de la string1 en la del buffer, luego concatena ambas strings. Por ultimo imprime el valor en la dirección del buffer y libera la memoria que este ocupa en el programa. El resultado del programa es la cadena “Hola Mundo”

ESTRUCTURAS

1. Si, en el stack.
2. Formato en el cual se almacenan datos de más de un byte en un ordenador.
3. Little Endian, el lenguaje C utiliza el Little endian en el manejo de la informacion
4. El programa se realizo, imprimiendo las variables del cd.
5. ---



1. Intente probar el programa pero en un punto saca el error “Segmentation fault (core dumped)



1. Programa

