

Laboratorio 0

Programa Hola mundo en C y comandos Linux (No evaluado)

1 Instalar la máquina virtual

Descargue la MV de link que aparece en Canvas.

Descargue Virtualbox para su S.O.

Importe el archivo OVM desde su disco duro.

Proceso para computadores Apple con procesador ARM M1, M2...

<https://github.com/utmapp/UTM/issues/3050#issuecomment-1058404656>

2 Iniciar el entorno de programación

En el shell, lo mas probable es que Ud se encuentre en la carpeta raíz del usuario PBN. En esa carpeta se encuentran las otras carpetas base de ese usuario, similares a las que se encuentran en otros sistemas operativos: Documentos, descargas, escritorio, etc.

Dentro de la carpeta Documentos, cree una carpeta para la asignatura, y dentro de esta, una para clases, una para las tareas y una para los laboratorios.

Ubíquese en el la carpeta laboratorio.

3 Crear documento

Usando el comando **touch lab0.1.c** permite crear un documento vacío, que luego será usado para digitar el código fuente.

4 Abrir el editor

Usando el comando **gedit lab0.1.c &** se abre una nueva ventana, que corresponde al programa GNU edit. Es un editor bastante simple, pero que cumple con lo necesario para este curso.

Nota con el comando anterior: Se usa un signo and (&) al final del comando para que se ejecute la inicialización del editor, pero que se abra en un proceso independiente, volviendo el control al shell.

5 Programa Hola mundo

Digite el siguiente programa en el editor

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

```
int main() {  
    printf("Hola mundo\n");  
    return 0;  
}
```

Para guardar el documento, se debe presionar control + s

6 Compilación

Una vez guardado el documento, se puede compilar. En el shell se ingresa el comando **gcc lab.0.1.c -o lab.0.1**

Si tod funciona bien, sin errores, no se va a ver ninguna salida en el shell. Si aparece algún mensaje, debe revisar donde se produjo el error y corregir.

7 Ejecución

Si la compilación terminó sin defectos, en su carpeta laboratorio debe haber dos archivos: El archivo fuente (.c) y su archivo ejecutable. Este último de diferencia del anterior porque es verde.

Para iniciar el programa, se debe escribir el comando **./lab.0.1**

El indicador **./** es para indicar que lo debe buscar en la carpeta actual.

Si todo funciona bien, se va a ver la palabra Hola mundo en la pantalla. En rigor, puedo volver a ejecutar su aplicación las veces que sea necesario, sin necesidad de volver a compilar. Sin embargo, si realiza cambios en el código fuente, se hace necesario volver a compilar y revisar por errores.

8 Comandos de consola (BASH)

Linux cuenta con una gran cantidad de comandos útiles para cuando no se cuenta con una interfaz de usuario gráfica. Veremos algunos de ellos. La idea es que se familiarice con la consola de Linux, utilizando esta serie de comandos.

8.1 Navegación por el sistema de archivos

- **pwd** Muestra el directorio actual
- **ls** [opciones] Muestra el contenido actual del directorio. Algunas opciones son:
 - o -R para listar recursivamente, entrando a cada subdirectorio
 - o -l para ver una lista detallada
 - o -r para invertir el orden
 - o -S para ordenar por tamaño
 - o -t para ordenar por fecha
- **cd** dir: cambia al directorio "dir"
- **mkdir** dir: Crea el directorio "dir"

8.2 Redirección de input y output

En la consola, el input por defecto es el teclado, y el output por defecto es la pantalla. Sin embargo, se pueden modificar por archivos, tanto para la entrada como la salida.

- `cmd > archivo`: hace que el output de `cmd` vaya al archivo. Si existe, se reescribe
- `cmd >> archivo`: Hace que el output de `cmd` va al archivo. Si existe, se adjunta al final
- `cmd < archivo`: Hace que se le pasen a `cmd` como input lo contenido en el archivo

8.3 Búsqueda de archivos y contenido

Comandos para el contenido de un archivo

- **tail** [opciones] archivo[s]: Muestra las 10 últimas líneas del archivo. Si se especifican mas de un archivo se muestran las 10 últimas líneas de cada archivo. Esto es muy útil para archivos log. Opciones:
 - o `-f` deja comando tail latente, de manera que muestre mas líneas en la medida que se van generando
 - o `-n num`, muestra las últimas num líneas
- **head** [opciones] archivo[s]: Similar a tail, pero muestra las 10 filas del principio del archivo. Opciones.
 - o `-f` deja comando head latente, de manera que muestre mas líneas en la medida que se van generando
 - o `-n num`, muestra las primeras líneas
- **cat** [archivo[s]]: Muestra en pantalla el o los archivos especificados
- **grep** [opciones] patrón [archivo[s]]: Busca en el archivo indicado el patrón. Si lo encuentra, muestra esa línea.

8.4 Otros comandos

- **rm** [opciones] [archivos]: Borra archivos o directorios. En el último caso, el directorio debe estar vacío. Opciones:
 - o `-r` borrado recursivo, incluso si el directorio no está vacío.
 - o `-f` Forzar el borrado
 - o `-v` Verbosamente, mostrar el detalle
- **cp** [opciones] [original][destino] Copia ruta y archivo original al destino. Opciones:
 - o `-r` Recursivamente
 - o `-v` Verbosamente
- **mv** [opciones] [original][destino] Mueve ruta y archivo original al destino. Opciones:
 - o `-r` Recursivamente
 - o `-f` Forzar la movida de archivos
 - o `-v` Verbosamente
- **tar** [opciones][archivos]: Comprime o descomprime archivos. Opciones:
 - o `-c` nombre, crea un nuevo comprimido con nombre
 - o `-x` Descomprime el archivo
 - o `-f` nombre: usa el archivo nombre
 - o `-v` Berbosamente
 - o `-j` Usa el compresor bzip2
 - o `-z` Usa el compresor gzip