

# Laboratorio 2: Herramientas para Desarrollo de Programas (2); Introducción a CentOS (Linux)

## Objetivos

- Conocer el sistema operativo CentOS.
- Conocer y utilizar algunos comandos básicos de Linux.
- Crear, compilar y correr programas en C.

**Duración:** 1 sesión

## Primera Parte: Conociendo el sistema operativo CentOS; Comandos básicos de Linux

Las computadoras de los laboratorios pueden arrancar en Windows o en CentOS, que es una distribución de Linux. Preste atención a las instrucciones para arrancarlas en CentOS.

1. Tome unos minutos para explorar el sistema operativo CentOS. En particular, revise el menú “**Aplicaciones**” en la esquina superior izquierda del escritorio.
2. Abra un explorador de archivos (*Aplicaciones > Accesorios > Archivos*) y busque el folder Documentos. En dicho folder, cree un folder con su nombre (clic derecho → Carpeta nueva). De preferencia, no incluya espacios en blanco en el nombre del folder (ej. Luis\_Rivera).
3. Abra el editor de texto **gedit**, que es el editor predeterminado de CentOS (*Aplicaciones > Accesorios > Editor de textos*). Guarde un archivo con extensión .c en el folder que creó en el inciso anterior. Procure que el nombre del archivo no contenga espacios en blanco (esto no es obligación, pero es conveniente).

Como en Windows y otros sistemas operativos, la interfaz gráfica permite configurar y hacer muchas cosas. En sistemas embebidos, muchas veces no es posible utilizar una interfaz gráfica, por lo que debemos aprender a usar la consola (terminal, línea de comandos), la cual es basada en texto. La consola permite navegar por el sistema de archivos, compilar y ejecutar programas, configurar el dispositivo, y muchas cosas más.

4. Abra una consola (*Aplicaciones > Herramientas del sistema*). A continuación, probará varios comandos básicos. Siga las instrucciones del catedrático. [Después de la práctica, en su reporte, deberá describir brevemente para qué sirven los comandos.](#) Puede buscar los comandos en Internet, o puede usar el comando **man** (manual), que accede a descripciones de los otros comandos.

**man** comando

Como ejemplo, pruebe: **man ls**

**ls** [opciones]

Pruebe: **ls**, **ls -l**, **ls -la**

Las opciones se pueden ver en la documentación. Son opcionales.

**pwd**

**cd** ruta\_del\_directorio                      Pruebe: **cd** Documentos, **cd** .. , **cd** ruta/a/su/folder, **cd**

**mkdir** [opciones] nombre                      Pruebe crear un subfolder dentro de su folder.

**cp** [opciones] fuente destino                      Pruebe copiar el archivo creado en el inciso 3 a Documentos.

**rm** [opciones] nombre                      Pruebe eliminar el archivo que copió en Documentos, y pruebe eliminar el subfolder creado con mkdir.

**mv** [opciones] fuente destino                      Pruebe mover el archivo que creó en el inciso 3 al Escritorio, y luego muévelo de regreso al folder con su nombre.

**clear**

**ps** [opciones]                                      Pruebe: **ps**, **ps ax**

**top**

**who** [opciones]                                      Pruebe el comando sin opciones, y pruébelo con alguna opción.

**date** [opciones]                                      Pruebe el comando sin opciones, y pruébelo con alguna opción.

**ifconfig** [opciones]                                      Pruebe el comando sin opciones, y pruébelo con alguna opción.

## **Segunda Parte: Visual Studio Code, compilación y ejecución de programas en C, en Linux**

Es posible que Visual Studio Code (VSC) no esté instalado en CentOS. Por lo tanto, lo primero que hará será instalarlo, y lo hará desde la línea de comandos.

1. Abra una consola. Para poder hacer la instalación, debe cambiar a “super usuario” (*super user*). Para ello, ingrese el comando:

**su**

El instructor ingresará la contraseña en cada estación. Note que el usuario ahora es “root” en lugar de “lab”.

2. Abra el explorador Firefox. Debería encontrar un acceso directo en el Escritorio (aunque no tiene el ícono usual de Firefox). Vaya a la siguiente página:

<https://code.visualstudio.com/docs/setup/linux>

Busque la sección “RHEL, Fedora, and CentOS based distributions”.

3. En la consola, ingrese las líneas del primer recuadro de la sección (cópielas y péguelas).
4. Luego, ingrese:

**sudo yum check-update**

Si le aparecen mensajes solicitando que ingrese y/n (yes/no) o s/n (sí/no), ingrese y o s, según sea el caso, para aceptar. El proceso de actualización puede tomar algunos minutos.

5. Luego, ingrese:

**sudo yum install code**

De nuevo, es posible que le aparezcan mensajes solicitando confirmación (y/n o s/n). Ingrese y o s para aceptar. La descarga e instalación también puede tomar algunos minutos. Si todo salió bien, debería aparecer el mensaje ¡Listo!

6. Cuando termine la instalación, debe regresar al usuario lab. Para ello, ingrese:

**exit**

Verifique que el usuario es nuevamente “lab”.

7. Para abrir VSC, ingrese:

**code**

Es posible que le aparezca una ventana con un mensaje sobre introducir contraseña para desbloquear algo. Ingrese: IEMTBM305 o IEMTBM306

Si no funciona ninguna de las dos, dele clic en cancelar.

También es posible que le aparezcan algunos mensajes de advertencia propiamente de VSC. Ignórelos/ciérrelos.

8. Finalmente, vaya a la opción de Extensiones e instale: **C/C++ Extension Pack** y **Code Runner**.

9. En VSC, abra el folder que creó en el inciso 2 de la Primera Parte, y el archivo .c que creó en el inciso 3 de esa parte. Edite el archivo para crear un programa “**Hello World**” como en el laboratorio 1. Corra el programa. Los pasos son iguales a los realizados en Windows o macOS. Se recomienda cambiar la configuración para que la ejecución de los programas al hacer clic en el botón “Run Code” se haga directamente en la terminal, no dentro de la pestaña Output.

10. Como vio en el laboratorio 1, los programas se pueden correr dentro de la terminal integrada en VSC, pero también se pueden correr desde una consola aparte. Abra una consola y utilice el comando **cd** para ir al directorio donde creó su programa. El ejecutable debería estar en dicho folder (verifíquelo con **ls**). Por defecto, el nombre del ejecutable creado desde VSC será el mismo que el nombre del programa, aunque eso se puede cambiar. Corra el programa:

**./nombre\_ejecutable**

11. Desde la consola externa a VCS, cree un nuevo folder (llámelo, por ejemplo, Lab2) y copie el archivo fuente (.c) al folder recién creado. Vaya al nuevo folder. Use el comando **mv** para cambiar el nombre del archivo (llámelo, por ejemplo, lab2.c). Lo anterior se puede hacer desde el explorador de archivos, pero hágalo desde la consola, para practicar. Abra el archivo con el editor de texto **gedit**. Puede hacerlo desde el explorador de archivos (doble clic o clic derecho → *Abrir con Editor de textos*), pero también puede abrirlo desde la consola, ingresando:

**gedit lab2.c &**

El símbolo & se usa para mandar el proceso al trasfondo. No es necesario, pero eso permite continuar en control de la consola.

12. Compile y corra el programa desde la consola. Recuerde que para compilar se ingresa:

**gcc archivo\_fuente.c -o nombre\_ejecutable**

13. Modifique el programa para que haga lo descrito a continuación. Puede editar su programa desde VCS o desde **gedit**, como prefiera.
  - a. Inicialice una variable de tipo entero con el valor de cero.
  - b. Utilice un bucle “for” (“for loop”) que realice 20 iteraciones. En cada iteración se debe incrementar la variable entera en uno, y se debe desplegar el valor en la consola. Luego de desplegar el valor, el programa debe “dormir” por 1 segundo (utilice **sleep**).
  - c. Luego de que el bucle termine, el programa debe desplegar un mensaje de despedida, antes de finalizar.
14. Abra tres consolas. Pueden ser dos externas a VCS y la integrada, o tres externas a VCS. En dos de ellas, corra su programa simultáneamente (es posible correr múltiples instancias de un mismo programa al mismo tiempo). En la tercera consola, cuando los programas estén corriendo, ingrese el comando: **ps ax**  
**En su reporte, deberá describir lo que observe.**
15. Es posible terminar un programa antes de que este finalice su ejecución normal. Corra su programa en una de las consolas y cuando esté en ejecución, presione **Ctrl c**.
16. Asegúrese de guardar su programa. **Lo deberá subir a Canvas**. Será conveniente mantener un folder con todas sus prácticas en Google Drive, OneDrive, etc. Puede abrir su cuenta de Drive (OneDrive, etc.) desde el navegador de Internet en CentOS. **Se sugiere usar Firefox**. Otra muy buena opción es tener un repositorio de GitHub, y guardar sus programas allí. También puede guardar sus archivos en una memoria USB. Lo importante es tener una copia de respaldo de sus archivos.

**Importante: NUNCA apague la computadora.** Si quiere regresar a Windows, basta con reiniciar la computadora. Por defecto, esta arrancará en Windows.

### **Evaluación:**

Deberá entregar un .pdf con **las descripciones solicitadas en la guía**. Identifique claramente los incisos (no todos los incisos requieren respuesta). Además, deberá entregar su archivo fuente del programa creado en el inciso 13 de la Segunda Parte (.c). **Ambos archivos deben ser subidos a Canvas, a más tardar la noche previa al siguiente laboratorio.**

Asistencia y trabajo en el laboratorio:	40%
Guía completada (.pdf):	30%
Programa (.c):	30%

### **Material de Apoyo**

1. Sobre la terminal de Linux (hay muchas páginas y libros con información al respecto):  
<https://www.technodyan.com/que-es-la-terminal-de-linux/>
2. En Canvas se publicarán manuales, programas de ejemplo, ejecutables y otra documentación que serán útiles/necesarios a lo largo del semestre.