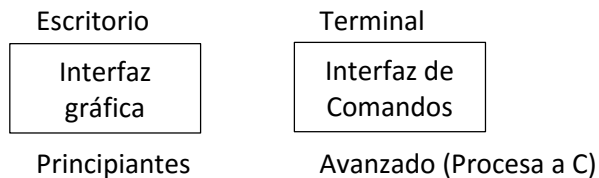


Resumen de Ubuntu

Ubuntu (Proviene de la [ética homónima](#), en la que se habla de la existencia de uno mismo como cooperación de los demás). Es un sistema operativo basado en [GNU/Linux](#) y que se distribuye como [software libre](#), el cual incluye su propio entorno de escritorio denominado Unity.

Los sistemas Operativos se pueden operar desde dos ventanas:



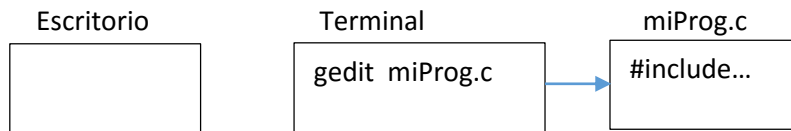
El computador arranca en la ventana **Escritorio**, de allí se abre la ventana **Terminal**; en la cual aparece el **prompt** (apuntador, entrada) para escribir los comandos. Ambas ventanas son independientes operacionalmente; pero lo que se hace en una se refleja en la otra; por ejemplo, si se crea una carpeta en **Terminal**, se refleja en **Escritorio**.

Comandos de **Terminal** más usuales

Comando	Descripción	Ejemplo
ls	Lista las carpetas y archivos contenidos en la carpeta actual	ls
cd miCarpeta	Cambia a la subcarpeta miCarpeta	cd Escritorio cd carpeta1/carpeta11
cd ..	Cambia a la supercarpeta	cd ..
mkdir miCarpeta	Crea miCarpeta en la carpeta actual	mkdir 01Clase
rm archivoCarp	Borra archivo o carpeta vacía	rm archi1 rm miCarpeta borra carpeta vacía rm -r miCarpeta borra carpeta llena rm -f miCarpeta no pide confirmación antes de borrar rm -f -r miCarpeta
cp arch1 arch2	Copia arch1 con el nombre arch2 (opcional); puedes ubicarlo en una carpeta diferente	cp miArch /home/miArch cp miArch /home/miArch2
mv arch1 arch2	Mueve arch1 con el nombre arch2 (opcional); puedes ubicarlo en una carpeta diferente	mv miArch /home/ mv miArch /home/miArch2 mv miArch1 miArch2 (renombra)
clear	Limpia la ventana Terminal	Clear
^c	Aborta, cancela un comando en ejecución; por ejemplo cancela un programa en loop.	^c
gedit miProg.c	Activa el editor gedit para el archivo miProg.c	gedit 01Prog.c Guardar el archivo en la carpeta actual.
gcc miArchivo.c	Compila un miArchivo de C	gcc 01Prog.c : compila y genera el programa a.out gcc 01Prog.c -o 01Prog : compila y genera el programa 01Prog
./miProg	Ejecuta a miProg (Programa objeto)	./01Prog

Buscar en google: "Ubuntu comandos"

Podemos abrir varias ventanas al mismo tiempo, por ejemplo:



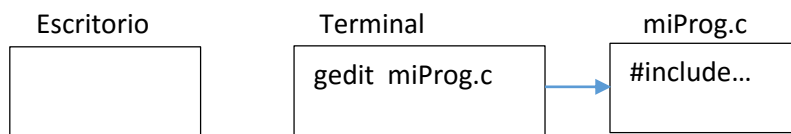
Navegaciones

Entre ventanas: Alt+tab +tab +tab ... (mantener presionada la tecla **Alt** con el pulgar izquierdo y repicar tab tab ... con el índice izquierdo) para cambiar de ventana.

Entre comandos ya ejecutados: Si ejecutó los comandos c1, c2, c3, c4; puede activar en el **prompt** los comandos c4, c3, c2, ... replicando la tecla **flecha arriba** (una a la vez); si replica la tecla **flecha abajo**, muestra los comandos en orden inverso.

Esta navegación es muy frecuente para repetir comandos anteriores; inclusive se muestra, modifica y ejecuta un comando.

Atento: Si abre las ventanas:



Las ventanas **Terminal** y **miProg.c** no son independientes operacionalmente. No puede ejecutar comandos en **Terminal** ya que el comando gedit se está ejecutando. Para ejecutar comandos debe **guardar** y **cerrar** miProg.c. Esto es muy incómodo; peor aún si ejecuta **^c** perderá todo lo que editó.

Para superar esta dificultad (Hacer que las ventanas **Terminal** y **miProg.c** operen independientemente), ejecute: gedit miProg.c& . Cada vez que modifique **miProg.c**; guarde (ya no es necesario cerrar); navegue a **Terminal** y compile; esto le permite corregir errores de compilación con facilidad.

Corregir errores de compilación

Si el programa fuente tiene errores de sintaxis, al momento de compilar se listan los errores indicando la línea y columna donde los detectó el compilador; por ejemplo:

Línea Instrucción o directiva

04 x = x + 1 // me olvidé “;”

05 x = x + 2;

El error dirá:

.....05:08 missing “.”, “..”, “;” or “..” No reporta el error en la línea 4, porque como no hay “;” en la línea 4, el compilador continúa con la instrucción en la línea 5.

Los errores de lógica no los detecta el compilador, por ejemplo:

Línea Instrucción o directiva

04 x = x + 1; // en lugar de x = x + 2; no será detectado, ya que no es error de sintaxis.