

Ayudantía 1 - Estructura de Datos y Algoritmos ELO320

Introducción a la línea de comandos de Linux y al lenguaje C

Felipe Vera A.

20 de marzo de 2014

1. Introduciendo a Linux

1.1. Iniciar Linux desde una máquina virtual

En cada máquina está instalada una distribución de Linux (*Ubuntu 12.04*) via la máquina virtual *Oracle VirtualBox*, lo que permite acceder a Linux desde una instalación con Windows.

Para acceder a Linux, se debe acceder al programa *Oracle VirtualBox*, seleccionar la máquina virtual *AyudantiaEDA* y luego hacer clic en *Iniciar*.

[Colocar un screenshot]

1.2. Usar la terminal

Se puede acceder a la terminal de Linux de dos maneras:

- En el menú en la parte superior de la pantalla, accediendo a **Aplicaciones** → **Accesorios** → **Terminal**.
- Presionando la combinación de teclas **Ctrl + Alt + T**.

La terminal comienza automáticamente en la carpeta del usuario `\home\alumno`.

1.3. Comandos más usados

- **cd** *<carpeta>*: Cambiar la carpeta actual. Para ir a la carpeta superior, se usa `..` como *<carpeta>*.
- **rm** *<archivo>*: Borra un archivo. Se puede agregar los siguientes parámetros:
 - **-f**: (Forzar) No solicita confirmación antes de borrar el archivo.
 - **-r**: (Recursivo) Borra una carpeta y todos sus contenidos.
- **ls**: Entrega una lista con todos los archivos presentes en la carpeta actual. Si se quiere una lista con más detalles se puede usar el parámetro **-l**.
- **man** *<comando>*: Manual de referencia para algún comando del terminal, programa o función en C.
- **nano**: Es un editor de textos posible de operar desde el terminal. Se puede añadir el nombre de un archivo como parámetro para editarlo.
- **cat, less**: Visualiza el contenido de un archivo en el terminal.

1.4. Teclas

- **Tab**: Rellena un comando sin necesidad de escribirlo completamente.
- **Flechas** ↑ y ↓: Busca los comandos ingresados anteriormente en el terminal.
- **Ctrl + C**: Cierra el programa ejecutándose en el terminal.

2. Creando un programa en C

La gran diferencia entre C y *Python* es que C no es un lenguaje interpretado, es decir, el programa no cambia inmediatamente tras guardar los cambios en el código fuente, sino que es necesario *compilar* ese código para crear un programa.

No es necesario usar una IDE para crear un programa en C. Sólo se necesita un editor de textos para editar el código fuente (actualmente está disponible el editor *gedit*) y un compilador para crear el programa a partir del código. En este curso se usará *gcc*.

Sintaxis básica de gcc

- **gcc <fuente.c>**: Crea un ejecutable llamado *a.out* compilando el archivo fuente proporcionado como parámetro.
- **gcc <fuente.c> -o <ejecutable>**: Al pasar este argumento es posible definir el nombre del ejecutable de salida.

2.1. Incluir una librería

Para incluirla se usa la directiva `#include`. La librería `stdio.h` provee funciones para mostrar información en la consola, y debe colocarse antes de ocupar cualquier función de ella.

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

2.2. Ejecutar acciones

Al comenzar el programa, este llega a la función `main`. Úsela para programar las primeras acciones que tomará su programa al ser ejecutado. Comúnmente es de tipo `int` y recibe dos argumentos:

- `int argc`: Cantidad de argumentos proporcionados. El nombre del programa también cuenta como uno.
- `char **argv`: Valores que tienen estos argumentos.

```
int
main (int argc, char **argv)
{
    [...]
    return 0;
}
```

2.3. Ejecutar el programa

No se puede llamar al programa simplemente escribiendo su nombre. Debe anteponer un `./` al nombre de su programa para ejecutarlo. A continuación un ejemplo:

```
./a.out
```

3. Ejercicios

1. USO DEL TERMINAL: Cree una carpeta y un archivo fuente (con extensión .c) dentro de ella. Acceda a esta carpeta desde el terminal.
2. INTRODUCCIÓN A C: Dentro de ese archivo fuente, incluya la librería `stdio.h` y use la función `printf` para imprimir la cadena *Primer programa en EDA!*. Si tiene dudas sobre esta función, consulte la referencia de la de la Librería General GNU, localizada en el escritorio como *glibc.pdf*. Compile y ejecute el programa, de modo que el ejecutable de salida se llame `ayudantiaeda01`.
3. VARIABLES EN C: Cambie el programa y haga que imprima un número contenido en una variable. Luego haga que se imprima un mensaje 100 veces con dicho número, y que por cada vez que este mensaje se imprima, este número se incremente en 1.
4. FUNCIONES EN C: Modifique el programa anterior y haga que la rutina de mostrar el mensaje esté implementado en una función aparte que llamará dentro de `main`. El número mostrado en dicho mensaje deberá ser pasado como argumento a esta función.
5. ARGUMENTOS DE PROGRAMA, ARREGLOS: Haga que el programa imprima todos los argumentos que le fueron proporcionados, separados por una línea nueva (`\n`).

¡RECUERDE! Su ayudante puede responder las dudas que le surjan, ¡no se quede con ellas!

Si le interesa instalar una versión definitiva de alguna distribución de Linux en su computadora personal (lo que le ayudará mucho en el transcurso de este curso), puede preguntarle detalles a su ayudante.