

Trabajo práctico de Laboratorio 01

Introducción al shell del sistema GNU/LINUX



Introducción a la computación
Departamento de Ingeniería de Computadoras
Facultad de Informática - Universidad Nacional del Comahue



Lectura obligatoria:

- Apunte del shell de Linux: <http://pedco.uncoma.edu.ar/mod/resource/view.php?id=207175>
- Apunte introductorio a **BASH**: <http://pedco.uncoma.edu.ar/mod/resource/view.php?id=244968>
- Linux Man Pages Online: <https://linux.die.net/man/>

A continuación, se realizarán una serie de ejercicios para los cuales necesitará acceso a una computadora con el sistema operativo de tipo **UNIX** (como por ejemplo alguna distribución **GNU/Linux**) y las siguientes aplicaciones:

- Intérprete de comandos **BASH** y demás utilidades que se encuentran en la mayoría de las distribuciones de Linux.
- Un editor de texto en la interfaz de línea de comando o gráfica.

Para un tutorial sobre como realizar una conexión remota a los laboratorios de la facultad, diríjase al anexo de la página 7.

1. Entrando en calor con el *shell* BASH

El sistema operativo controla diferentes procesos de una computadora. Uno de ellos es el intérprete de comandos o *shell*. Este es un programa que permite al usuario interactuar con el Sistema Operativo. Permite iniciar (ejecutar) otros programas así como también tiene comandos propios que no necesitan de otros programas, como por ejemplo comandos para movernos en la estructura de directorios. En este práctico, utilizaremos el intérprete **BASH** (*Bourne-Again SHell*), muy popular en el mundo de Linux.

Dicho programa debe ser ejecutado en lo que llamamos Terminal o Consola, que es otro programa que nos permite interactuar mediante el teclado (ingresar caracteres) y ver los resultados de la ejecución de otros programas en la pantalla. Al iniciar una consola o terminal de textos, automáticamente se inicia el *shell* **BASH**, que es con el que el usuario realmente interactúa (más abajo se explica esto con un ejemplo).

1. Ingrese remotamente a los laboratorios siguiendo alguna de las opciones plateadas en el tutorial del anexo 3.
2. Una vez realizada la conexión, el *shell* **BASH** en la terminal espera que se escriban órdenes para ejecutar (comandos). En la terminal escriba el comando **hostname** y presione la tecla *enter* para verificar el nombre del equipo al que se conecto.

3. Ejecute el comando `who` para ver que usuarios están conectados actualmente al sistema.
4. Ejecute el comando `lscpu` para ver las características del *CPU* del sistema al que se ha conectado.
5. Ejecute el comando `man free` para leer el manual del comando `free`. Para salir, presione la tecla `q`.
6. Ejecute el comando `free` ¿Cómo puede hacer para que muestre la información en `GiB`? (ayuda: lea el manual)
7. Ejecute el comando `top` para ver los procesos que están ejecutando en el sistema de forma dinámica, así como información del uso del tiempo de *CPU* y memoria. Para salir, presione la tecla `q`.

2. Sistemas de archivos

El concepto de directorios (comúnmente llamados “carpetas”) y archivos es hoy en día familiar a los usuarios de computadoras. En general el usuario se maneja visualmente con el ratón y el Explorador de Archivos, utilizando estos para acceder a los directorios, abrir archivos, ejecutar programas, etc. Podemos pensar el sistema de archivos como una forma de organizar todo el contenido en el dispositivo de almacenamiento. Los archivos son datos concretos, utilizan espacio físico del dispositivo, mientras que los directorios sirven para organizar “lógicamente” las cosas, igual que una persona tiene estanterías y cajones en un escritorio para organizar sus papeles.

La figura 1 muestra un esquema de como se puede pensar en una estructura de directorios y archivos. `ic`, `parciales` y `imagenes` son directorios, mientras que `notas.tsv`, `p1.pdf`, `p2.pdf` y `gatito.jpg` son archivos.

Cuando se inicia el intérprete de comandos, éste se posiciona en una determinada carpeta, es decir, los comandos que se escriben afectarán directamente al contenido de dicha carpeta.

En general, al iniciar una consola de texto, junto con un intérprete de comandos, se posiciona al usuario en la carpeta personal o “**HOME**” (en Linux en general es “`/home/usuarioX`”). Desde aquí el usuario puede navegar por sus archivos y ejecutar comandos.

Además, existen dos formas de hacer referencia a los directorios y a los archivos en el intérprete, absoluta y relativa. Cuando utilizamos la forma absoluta se escribe todo el camino desde la carpeta inicial (en Linux se denota con el símbolo “`/`”) hasta la ubicación deseada. Ejemplos de uso absoluto: `/home/usuario/trabaj01`, `/etc`, `/tmp`, etc.

El modo relativo sirve para hacer referencia a un directorio que se encuentra “cerca” del directorio de trabajo actual en la jerarquía, como por ejemplo el directorio directamente superior o inferior.

Por ejemplo, las siguientes son direcciones relativas:

- `./` (el directorio actual)
- `../` (directorio inmediatamente superior)
- `./trabaj01` (directorio trabajo1 ubicado dentro del directorio actual)
- `../../archivoX` (archivo ubicado dos directorios más arriba en la jerarquía)

8. Ejecute el comando `pwd` para ver en que directorio se encuentra actualmente el shell.

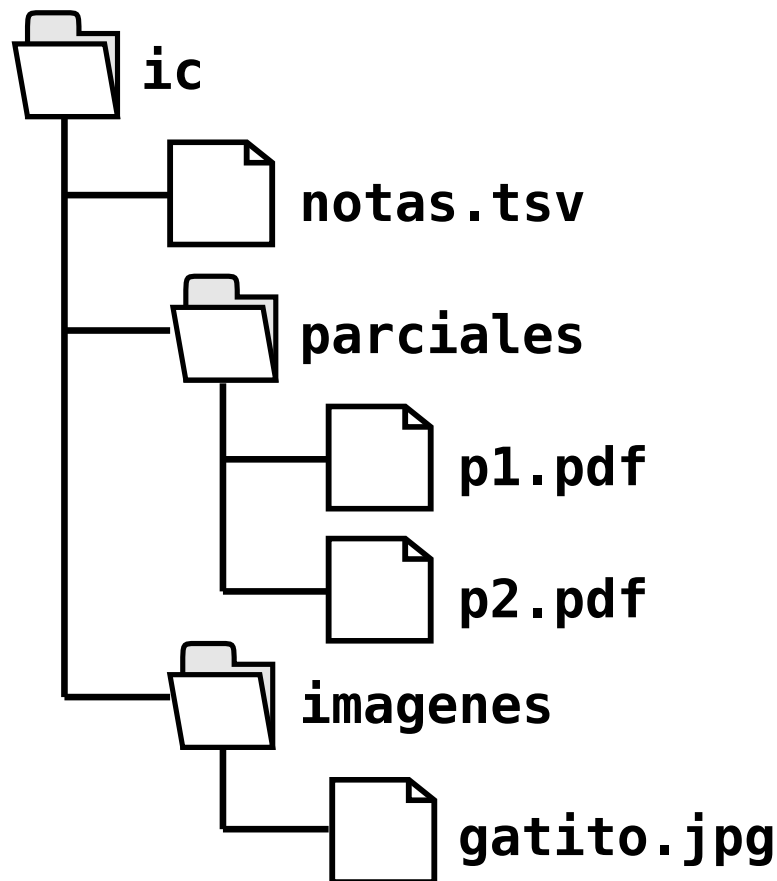


Figura 1: Árbol de directorios.

9. Ejecute el comando `ls` para listar los archivos que se encuentran en el directorio actual.
10. Para crear un archivo de texto ejecute el comando `nano archivo.txt`. Escriba el texto que le plazca. Para guardar presione `Ctrl+O`, presione `enter` para confirmar, y presione `Ctrl+X` para salir.
11. Ejecute el comando `ls` para verificar que el archivo fue creado exitosamente.
12. Ejecute el comando `cat archivo.txt` para mostrar el contenido del archivo en pantalla.
13. Ejecute la siguiente secuencia de comandos. Lo que se encuentra luego del caracter `#` es un comentario explicativo de que debe hacer el comando y sera ignorado por el shell, no es necesario tipearlo. IMPORTANTE: no olvide salir del editor `nano` antes de continuar ingresando los comandos subsecuentes.

Comandos y parámetros	Descripción
<code>mkdir e</code>	<code>#crear un directorio de nombre "e"</code>
<code>cd e</code>	<code>#cambiar el directorio actual a "e"</code>
<code>mkdir h1</code>	<code>#crear un directorio "h1"</code>
<code>mkdir h2</code>	<code>#crear un directorio "h2"</code>
<code>cd h1</code>	<code>#cambiar el directorio actual a "h1"</code>
<code>nano a.txt</code>	<code>#crear un archivo de texto "a.txt"</code>
<code>pwd</code>	<code>#imprime el directorio actual</code>
<code>cd ..</code>	<code>#cambiar el directorio actual al directorio padre</code>
<code>pwd</code>	<code>#imprime el directorio actual</code>
<code>cd ..</code>	<code>#cambiar el directorio actual al directorio padre</code>
<code>pwd</code>	<code>#imprime el directorio actual</code>
<code>ls</code>	<code>#listado del directorio actual</code>
<code>ls -R</code>	<code>#listado recursivo de los contenidos</code>
<code>mv e/h1/a.txt e/h2/a.txt</code>	<code>#mueve el archivo "a.txt" a otro directorio</code>
<code>rm e/h2/ab.txt</code>	<code>#eliminar el archivo "ab.txt", falla por que no existe</code>
<code>rm e/h2/a.txt</code>	<code>#eliminar el archivo "a.txt"</code>
<code>rmdir e/h1</code>	<code>#elimina el directorio "h1"</code>
<code>tree e</code>	<code>#muestra el árbol de directorios de "e".</code>

14. Realice un esquema visual como el de la **figura 1**, que muestre la estructura de directorios resultante de ejecutar la siguiente secuencia de comandos:

```
mkdir newdir
cd newdir
mkdir h1
mkdir h2
cd h1
nano README
cd ..
cd h2
mkdir h2 h3
cd h2
nano otroArchivo
cd ../../
nano a3.txt
```

Puede verificar que su esquema es correcto ejecutando el comando `tree newdir`.

15. Utilizando los comandos vistos en el listado, reproduzca la estructura de directorios y archivos de la figura 1, respetando mayúsculas y nombres de los archivos. Esta estructura será utilizada en los subsiguientes incisos.
16. Utilizando la jerarquía de directorios generados en el inciso anterior, posicione el intérprete en el directorio `imagenes` y realice las siguientes acciones:
- Elimine el archivo `notas.tsv` sin cambiar de directorio (repase la introducción de la sección 2).
 - Mueva los archivos `p1.pdf` y `p2.pdf` a la carpeta `imagenes`.

- c) Elimine la carpeta `parciales`.
- d) Cree una carpeta `nueva` dentro de la carpeta `ic`.
- e) Posicione el intérprete en el directorio `nueva`.
- f) Sin cambiar de directorio (es decir, sin utilizar el comando `cd`), listar el contenido del directorio `imagenes` usando rutas relativas.
- g) Sin cambiar de directorio(es decir, sin utilizar el comando `cd`), listar el contenido del directorio `imagenes` usando rutas absolutas. Para armar una dirección absoluta puede usar el comando `pwd` para conocer como se conforma el camino completo hasta la ubicación actual.
17. Explique las ventajas del uso de direcciones “relativas” a la hora de hacer referencia a otros archivos cercanos en la jerarquía al directorio de trabajo actual. Utilice un ejemplo.
18. Liste el contenido del directorio `imagenes` con el comando `ls -l camino_a_imagenes` (reemplace `camino_a_imagenes` por la ruta apropiada, `-l` es “L” minúscula). Analice la información que se muestra por pantalla, fecha de creación de los archivos, propietario de los archivos y tamaño de los archivos. Modifique alguno de los archivos en el editor de texto y vuelva a analizar la salida del comando `ls -l`(`-l` es “L” minúscula).
19. Ejecute los comando `cat`, `more` y `less` para visualizar el contenido del archivo `/etc/wgetrc` ¿Cual es la utilizad de estos programas? ¿Qué diferencias tienen?
- Teclas útiles:
- `cat`: no tiene.
 - `more`: “q” para salir, “barra espaciadora” avanzar página.
 - `less`: “q” para salir, “barra espaciadora” avanzar página, y “j”, ”k” para subir y bajar.
20. Retorne a su directorio **HOME**. Para esto ejecute: (**Atención:** el signo `$` a partir de ahora representa el *prompt* del *shell BASH*, no debe escribirlo al ingresar cada comando)

```
$ cd          # este es el comando para retornar a su HOME
$ pwd        # pwd le indica cuál es su directorio de trabajo actual
```

3. Administración de procesos

21. Cree un archivo con nombre `dance.sh` con el siguiente contenido. Verifique que ha copiado correctamente cada línea.

```
#!/bin/sh

while true; do
    printf "\t(>'-'>)\r"
    sleep 0.5
    printf "\t(<'-'<)\r"
    sleep 0.5
done
```

22. Agregue permisos de ejecución utilizando el comando `chmod +x dance.sh`.
23. Ejecute el programa ingresando en la terminal `./dance.sh`.
24. Abra otra ventana y utilizando los comandos `top` o `ps a` encuentre el `PID` del proceso del programa `dance.sh`. Termine el programa utilizando el comando `kill PID_DEL_PROCESO`.
25. Ejecute el comando `file` sobre el archivo `dance.sh` y `/bin/ls` ¿Cuál es la función del comando `file`?
26. ¿En qué tipo de lenguaje está escrito el programa `dance.sh`: compilado o interpretado? ¿Y el comando `ls`?

Anexo: Conexión remota

Para conectarnos remotamente tenemos varias opciones, pero para todas es crucial primero conocer el usuario de los laboratorios y su contraseña. Si tiene e-mail institucional, el usuario es el mismo que el de su correo (si tu correo es *peperina@est.fi.uncoma.edu.ar*, su usuario es *peperina*), y la contraseña inicial es tu número de DNI. Si no tiene cuenta de usuario, o quiere cambiar la contraseña, puede dirigirse a este sitio: <https://formulario.fi.uncoma.edu.ar/>.

Shell WEB

1. Abra en el navegador el sitio <https://aula-ssh.fi.uncoma.edu.ar/>.
2. Ingrese su usuario y presione la tecla *enter*.
3. Ingrese su contraseña. Por cuestiones de seguridad, no se mostrará ningún eco de las teclas presionadas (al presionar teclas no se mostrarán en pantalla). Al terminar, presione la tecla *enter*.
4. Si el usuario y contraseña son correctos, se iniciará el *shell* y esperará a que ingrese un comando.

PuTTY

1. Descargue e instale la última versión de *PuTTY*. Para cada sistema:

Microsoft Windows: diríjase a <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>. Descargue y ejecute la última versión.

Debian y derivados: En una terminal ingrese el comando `sudo apt update; sudo apt install putty` y presione la tecla *enter*.

2. Ejecute el programa *PuTTY*. En *Microsoft Windows* haga click sobre icono correspondiente, en sistemas *GNU/Linux* puede ejecutar el comando `putty` en la terminal. Al ejecutar correctamente, se mostrara una pantalla como la mostrada en la figura 2.
3. En el campo *Host Name* ingrese su nombre de usuario seguido de `cw@aularemota.fi.uncoma.edu.ar`. Por ejemplo si su nombre de usuario es *peperina*, debe ingresar `peperina@aularemota.fi.uncoma.edu.ar`.
4. En el campo *Port* ingrese el número 1200.
5. Para no tener que re ingresar los datos cada vez que se ejecuta el programa, seleccione el campo *Default Settings* y luego presione el botón *Save*.
6. Para conectarse al sistema remoto, presione el botón *Open*. Se abrirá una terminal virtual.
7. Ingrese su contraseña. Por cuestiones de seguridad, no se mostrará ningún eco de las teclas presionadas (al presionar teclas no se mostrarán en pantalla). Al terminar, presione la tecla *enter*.
8. Si el usuario y contraseña son correctos, se iniciará el *shell* y esperará a que ingrese un comando.

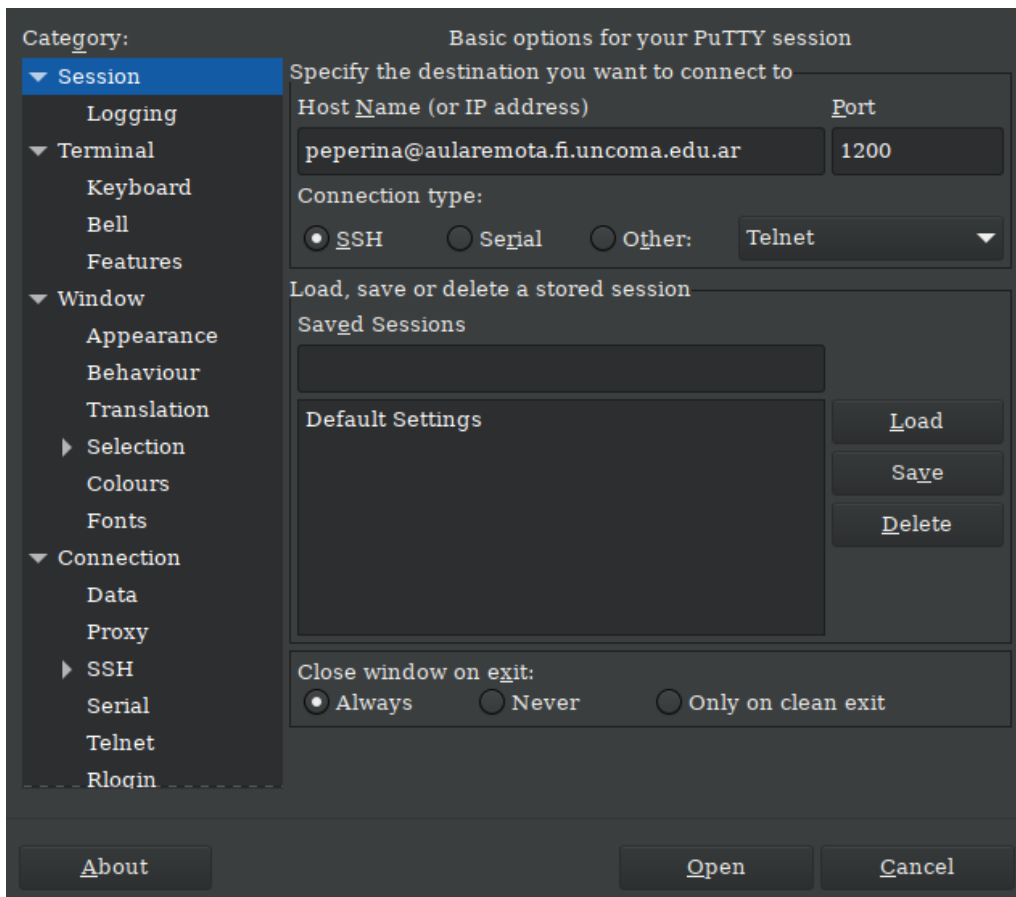


Figura 2: Ventana de inicio y configuración de *PuTTY*.

SSH

1. Abra una consola e ingrese el comando `ssh -p 1200 $USER@aularemota.fi.uncoma.edu.ar`, reemplazando *\$USER* por su usuario de los laboratorios. Presione la tecla *enter*.
2. Ingrese su contraseña. Por cuestiones de seguridad, no se mostrará ningún eco de las teclas presionadas (al presionar teclas no se mostrarán en pantalla). Al terminar, presione la tecla *enter*.
3. Si el usuario y contraseña son correctos, se iniciará el *shell* y esperará a que ingrese un comando.