Trabajo práctico de Laboratorio

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 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 8.

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 las personas usuarias del Sistema Operativo interactuar con el. Le permitirá iniciar (ejecutar) otros programas, así como también 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 la usuaria realmente interactúa (más abajo se explica esto con un ejemplo).

1. Ingrese remotamente a los laboratorios siguiendo alguna de las opciones planteadas en el tutorial del anexo de la página 8.

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 ¿Cuál es el nombre del equipo?

3. Ejecute el comando `who` para ver que los nombres de *login* de las personas que están conectadas en este momento al sistema ¿Cuántos hay conectadas actualmente?

4. Ejecute `last | tac` para ver la lista de últimas conexiones ¿Reconoce a alguna de las personas?
5. Ejecute el comando `lscpu` para ver las características del *CPU* del sistema al que se ha conectado ¿Cuántos núcleos tiene el sistema?
6. Ejecute el comando `man free` para leer el manual del comando `free`. Para salir, presione la tecla `q`.
7. Ejecute el comando `free` ¿Cómo puede hacer para que muestre la información en `GiB`? ¿Cuántos `GiB` tiene el sistema? *(ayuda: lea el manual)*
8. 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`. Identifique la cantidad de memoria utilizada actualmente.
9. Ejecute algunos de los siguientes juegos y elija su favorito:
   - `bastet`: Tetris® clone with "evil" block-choosing AI.
   - `empire`: the wargame of the century.
   - `greed`: eat a game field until you run out of moves.
   - `ninvaders`: ncurses version of space invaders.
   - `nsnake`: Classic snake game on the terminal.

2. Sistemas de archivos

El concepto de directorios (comúnmente llamados "carpetas") y archivos nos es hoy en día familiar. En general quienes utilizan el sistema se manejan 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 la persona usuaria en su carpeta personal o **"HOME"** (en Linux en general es ``/home/usuarioX''). Desde aquí la usuaria puede navegar por sus archivos y ejecutar comandos. Para saber cuál es el directorio en el que se encuentra, ejecute el comando `pwd` (*Print Working Directory*).

Además, existen dos formas de hacer referencia a los directorios y a los archivos en el intérprete, con rutas absolutas y relativas. 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/usuarioX/trabajo1`, `/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.

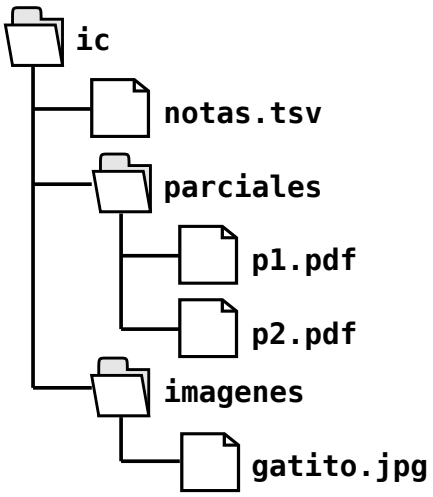

Figura 1: Árbol de directorios.

Por ejemplo, las siguientes son direcciones relativas:

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

10. Ejecute el comando `pwd` para ver en que directorio se encuentra actualmente el shell.
11. Ejecute el comando `ls` para listar los archivos que se encuentran en el directorio actual.
12. Para crear un archivo de texto ejecute el comando `nano archivo.txt`. Escriba el texto que le plazca (puede ser un poema, una frase sabia, o `ASCII Art`). Para guardar presione *Ctrl+O*, presione *enter* para confirmar, y presione *Ctrl+X* para salir.
13. Ejecute el comando `ls` para verificar que el archivo fue creado exitosamente.
14. Ejecute el comando `cat archivo.txt` para mostrar el contenido del archivo en pantalla.
15. Para mayor dramatismo, ejecute lo siguiente: `cat archivo.txt | cowsay -n | lolcat` (todos los comandos en la misma linea, separados por el carácter "`|`") ¿Qué hace cada uno de los comandos de la secuencia?
16. 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
mkdir e	#crear un directorio de nombre “e”
cd e	#cambiar el directorio actual a “e”
mkdir h1	#crear un directorio “h1”
mkdir h2	#crear un directorio “h2”
cd h1	#cambiar el directorio actual a “h1”
nano a.txt	#crear un archivo de texto “a.txt”
pwd	#imprime el directorio actual
cd ..	#cambiar el directorio actual al directorio padre
pwd	#imprime el directorio actual
cd ..	#cambiar el directorio actual al directorio padre
pwd	#imprime el directorio actual
ls	#listado del directorio actual
ls -R	#listado recursivo de los contenidos
mv e/h1/a.txt e/h2/a.txt	#mueve el archivo “a.txt” a otro directorio
rm e/h2/ab.txt	#eliminar el archivo “ab.txt”, falla por que no existe
rm e/h2/a.txt	#eliminar el archivo “a.txt”
rmdir e/h1	#elimina el directorio “h1”
tree e	#muestra el árbol de directorios de “e”.

17. Realice un esquema visual como el de la **figura 1** (ver página 3) 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
cd ..
```

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

18. 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.

19. Utilizando la jerarquía de directorios generados en el inciso anterior, posicione el intérprete en el directorio `imagenes` y realice las siguientes acciones:

    a) Elimine el archivo `notas.tsv` sin cambiar de directorio (repase la introducción de la sección 2).

b) 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.

20. 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.

21. Liste el contenido del directorio `imagenes` con el comando `ls -l` *`camino_a_imagenes`* (remplace *camino_a_imagenes* por la ruta apropiada, `-l` es "L" minuscula). 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" minuscula).

22. Ejecute los comando `cat`, `more` y `less` para visualizar el contenido del archivo `/etc/wgetrc` ¿Cuál 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.

23. 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

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

```
#!/bin/sh

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

25. Agregue permisos de ejecución utilizando el comando `chmod +x dance.sh`.

26. Ejecute el programa ingresando en la terminal `./dance.sh`.

27. Abra otra ventana (sin cerrar aquella donde se esta ejecutando el programa `dance.sh`) y utilizando el comando `ps a` encuentre el `PID` del proceso del programa `dance.sh`. Termine el programa utilizando el comando `kill` *`PID_DEL_PROCESO`*.

28. Ejecute el comando `file` sobre el archivo `dance.sh` y `/bin/ls` ¿Cuál es la función del comando `file`?

4. Lenguajes compilados e interpretados

4.1. Lenguajes interpretados

29. Cree un archivo con nombre `calcular.sh` con el siguiente contenido. Verifique que ha copiado correctamente cada linea.

```
#!/bin/sh
max=${1:-10}
num=0
for i in $(seq 1 "$max");do
    num=$((num + 3))
done
echo "El resultado de sumar 3 unas $max veces es: $num"
```

30. Verifique que el contenido del archivo se almaceno correctamente ejecutando el comando `cat calcular.sh`.

31. Agregue permisos de ejecución utilizando el comando `chmod +x calcular.sh`. Mida el tiempo de ejecución para 100, 100 000 y 1 000 000 utilizando el comando `time ./calcular.sh $NUM` (remplace `$NUM` por cada uno de los valores).

4.2. Lenguajes compilados

32. Cree un archivo con nombre `calcular.c` con el siguiente contenido. Verifique que ha copiado correctamente cada linea.

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

int main(int carg, char **varg)
{
    uint64_t max = 10, i, num = 0;
    if (carg > 1) {
        max = atoi(varg[1]);
    }
```

```
    for (i = 0; i < max; i++) {
        num = num + 3;
    }
    printf("El resultado de sumar 3 unas %d veces es: %d\n", max, num);
    return 0;
}
```

33. Verifique que el contenido del archivo se almaceno correctamente ejecutando el comando `cat calcular.c`.

34. Cree un archivo ensamblador optimizado con el comando `gcc calcular.c -S -O3` (`-O` es la letra "O" mayúscula). Inspeccione el contenido del archivo `calcular.s`.

35. Cree un archivo ejecutable con el comando `gcc calcular.s -o calcular` (`-o` es la letra "o" minuscula). Ejecute el comando `file calcular` ¿Qué información retorna el comando?

36. Mida el tiempo de ejecución para 100, 100 000 y 1 000 000 utilizando el comando `time ./calcular $NUM` (remplace `$NUM` por cada uno delos valores).

4.3. Comparando resultados

37. ¿Cuál de los dos ejecutables fue más rápido?

38. Modifique ambos programas para que sumen el número 42 en lugar de 3. Verifique que el cambio se realizo con éxito sumando 100 veces.

39. ¿Cuál considera que fue más simple de modificar?

40. Indique en sus palabras cuales son las ventajas y desventajas de los lenguajes compilados y los interpretados.

5. Redes de computadoras

5.1. Direcciones *IP* y enrutamiento

41. Utilice el comando `dig` para obtener la dirección *IP* del servidor cuyo nombre de dominio es "`uncoma.edu.ar`".

42. Verifique que sus mensajes pueden llegar hasta el servidor, usando el comando `ping -n uncoma.edu.ar` ¿Qué otra información provee el comando?

43. ¿Existe alguna diferencia entre ejecutar el comando `ping` utilizando la dirección *IP* en lugar del nombre de dominio del servidor? ¿Cuál es la utilidad de los nombres de dominio?

44. El comando `traceroute` permite obtener información sobre los routers por los que pasaran los paquetes en su camino hasta el servidor destino ¿Cuántos nodos hay entre su computadora y el servidor de "`uncoma.edu.ar`"?

Anexo: Conexión remota

Para conectarnos remotamente tenemos varias opciones, pero para todas es crucial primero conocer nombre de *login* de los laboratorios y su contraseña. Si tiene e-mail institucional, el nombre de *login* es el mismo que el de su correo (si tu correo es *peperina@est.fi.uncoma.edu.ar*, su nombre de *login* es *peperina*), y la contraseña inicial es tu número de DNI. Si no tiene cuenta en el sistema, 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 nombre de *login* 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 nombre de *login* 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 derribados: 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 *login* seguido de `@aula-ssh.fi.uncoma.edu.ar`. Por ejemplo si su nombre de *login* es *peperina*, debe ingresar `peperina@aula-ssh.fi.uncoma.edu.ar`.
4. En el campo *Port* ingrese el número `60173`.
5. Para no tener que re ingresar los datos cada ves 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 terminar 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 nombre de *login* y contraseña son correctos, se iniciará el *shell* y esperará a que ingrese un comando.

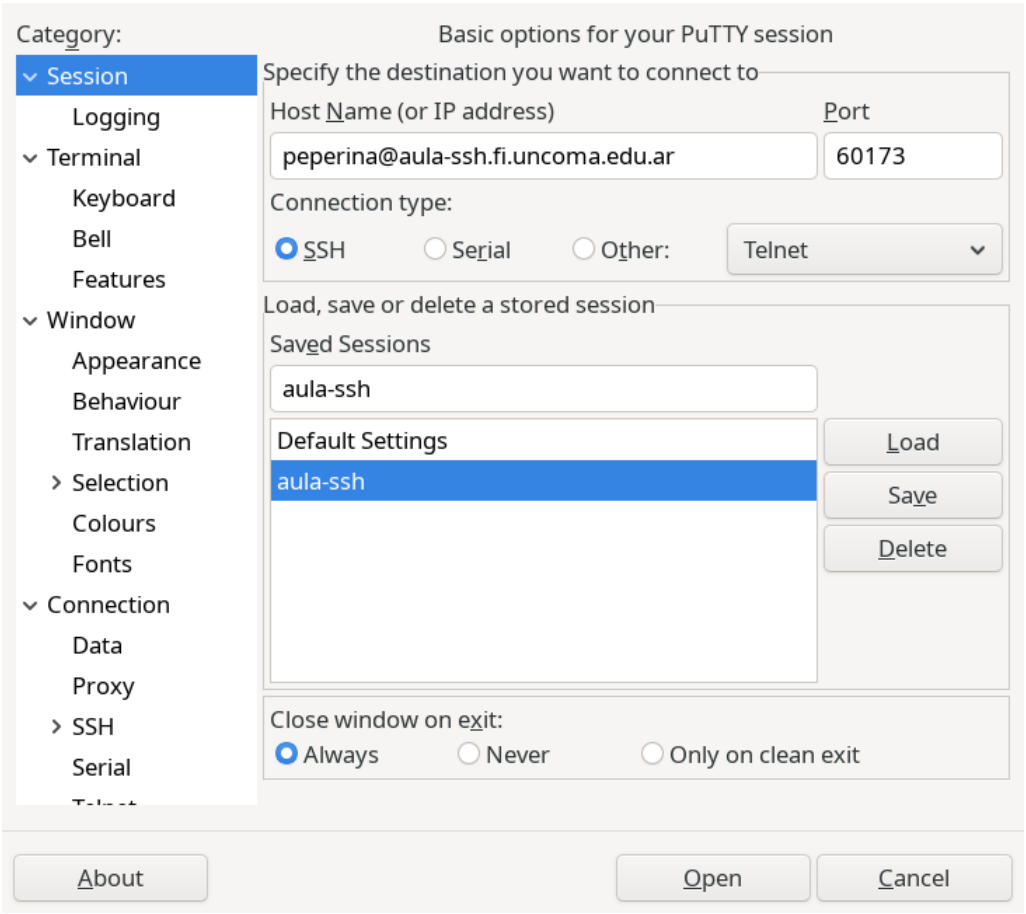

Figura 2: Ventana de inició y configuración de *PuTTY*.

SSH

1. Abra una consola e ingrese el comando `ssh -p 60173 $USER@aula-ssh.fi.uncoma.edu.ar`, remplazando *$USER* por su nombre de *login* 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 nombre de *login* y contraseña son correctos, se iniciará el *shell* y esperará a que ingrese un comando.