Algoritmos y Estructuras de Datos I

Guillaume Hoffmann, Walter Alini, Martin Dominguez

Interfaz gráfica: basicos

Apretar juntas las teclas ALT+F2 para ejecutar:

- `gedit`
- `firefox`
- `gnome-terminal`

Probar ALT+TAB para cambiar de ventana.

Si su sistema de ventanas no permite hacer ALT+TAB, desconectarse, y conectarse con otro sistema.

Probar F11 para maximizar la ventanas.

Manejo de la consola

Ejecutar la consola (`gnome-terminal`) y probar:

- tipear nombres de comando seguido de la tecla "Enter":
  - `ls`
  - `date`
  - `whoami`
- limpiar terminal: **CTRL+L**
- cerrar terminal: `exit` o **CTRL+D**

A partir de ahora trabajamos dentro de una consola.

Árbol de directorios

Los archivos están organizados en un árbol de directorios (o carpetas), es decir, una jerarquía de directorios y subdirectorios destinados a fines específicos.

Cuando se abre un terminal, estamos dentro de un directorio que es usualmente el directorio personal del usuario, indicada por el símbolo ~.

Para saber la ruta del directorio actual:

```
pwd  # Print Working Directory
```

Probar y comparar con sus vecinos.

Como leer una ruta

```
$ pwd
/home/guillaume
```

`/home/guillaume` : camino de mi carpeta actual.

`/home` : camino de la carpeta madre.

`/` : raiz del árbol de directorios de mi computadora.

```
$ ls
a b c
```

`/home/guillaume/a` : ruta de una carpeta hija.

`/home/guillaume/b` : también.

`/home/guillaume/c` : también.

Crear, moverse, mover, suprimir

Probar algunos commands `ls`, `mkdir`, `cd`, `mv`, `rm`:

```
ls                # listar los archivos de la carpeta actual
ls -l             # lo mismo, con más datos
ls -lh            # con tamaños de archivos leibles

mkdir dir1
mkdir dir1/subdir1

cd dir1           # moverse a una carpeta
cd subdir1
pwd

cd ..             # moverse a la carpeta madre
cd ..

mv dir1 carpeta1  # mover/renombrar carpeta

rm -r carpeta1    # suprimir (ReMove) carpeta
```

Copiar, usar la estrella

Creemos unos archivos vacíos:

```
touch  a.hs  b.hs  c.hs  x.c  y.c
```

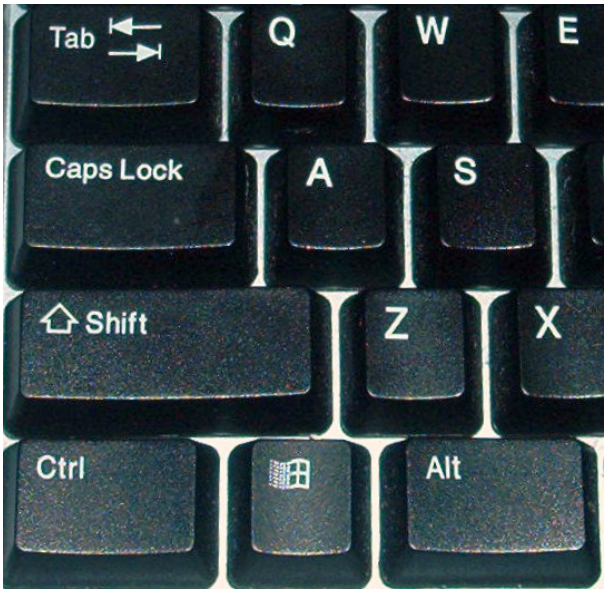
Probar:

```
ls *.hs
ls *.c
mkdir haskell
cp *.hs haskell/
rm *.hs
ls
```

Evitar errores y ganar tiempo

Apretar la tecla TAB (↹) para completar los nombres:

```
$ touch NombreLargisimo1111111
$ cd Nom                        # apretar TAB
$ cd NombreLargisimo1111111
```



Más operaciones

Con carpetas

```
mkdir nombre_carpeta              # crear la carpeta nombre_carpeta
mv  nombre_carpeta  nuevo_nombre  # renombrar
cp -r  nuevo_nombre  salvaguarda  # copiar la carpeta (necesita -r)
```

Con archivos:

```
mv   nombre_archivo   nuevo_nombre
cp   nuevo_nombre     salvaguarda
```

Copiar o mover un o varios archivos a una carpeta:

```
cp    a1 a2 a3    carpeta_destino/
mv    a1 a2 a3    carpeta_destino/
```

Chequear el uso del espacio

```
du -hs # imprime cúanto espacio ocupa la carpeta actual
df -h  # imprime el espacio ocupado en cada disco del sistema
```

¿Cúal es el tamaño de su disco?

Ver el contenido de un archivo de texto

Bajémonos la página wiki de la materia y mirémosla con `less`:

```
wget http://tinyurl.com/algo1-2013-2  # bajar la página
ls
less al                               # ahora apretar TAB
less algo1-2013-2
```

Probar estas teclas dentro de `less`:

flechas	: moverse
`q`	: salir
`/Corrección`	: buscar "Corrección" en el documento

`n` : saltar al próximo elemento encontrado

`N` : saltar al previo elemento encontrado

Comprimir/descomprimir

```
$ mkdir proy1
$ touch proy1/p proy1/q proy1/r proy1/s
$ # ahora creamos un archivo comprimido zip
$ zip -r MiSalvaGuarda.zip proy1/
  adding: proy1/ (stored 0%)
  adding: proy1/q (stored 0%)
  adding: proy1/p (stored 0%)
  adding: proy1/s (stored 0%)
  adding: proy1/r (stored 0%)
$ ls *.zip
MiSalvaGuarda.zip
$ rm -r proy1
$ # ahora descomprimimos nuestro archivo zip
$ unzip MiSalvaGuarda.zip
Archive:  MiSalvaGuarda.zip
   creating: proy1/
 extracting: proy1/q
 extracting: proy1/p
 extracting: proy1/s
 extracting: proy1/r
$
```

Configurar el editor de texto por defecto

Editar el archivo `~/.bashrc`. Si no saben:

```
nano ~/.bashrc
```

Agregarle la línea siguiente:

```
export EDITOR=gedit
```

O:

```
export EDITOR=nano
```

O:

```
export EDITOR=vim
```

etc.

Por ejemplo `ghci` se fija en la variable EDITOR a la hora de llamar un editor de texto.

Test con `ghci`

Chequeen con el interprete `ghci` que el editor especificado se ejecuta:

```
:e proyecto1.hs
```

Volver a editar el mismo archivo:

```
:e
```

Ejemplos de editores

- Textuales, se ejecutan en un terminal sin abrir ventana nueva:
  - `nano`
  - `mcedit`
  - `vim`
  - `emacs`
- Gráficos, se ejecutan en su propia ventana:
  - `gedit`
  - `kate`

Es importante conocer unos atajos de teclado para ser más eficiente

Atajos de editores gráficos (`gedit`...)

CTRL+O	abrir archivo existente
CTRL+S	guardar
CTRL+Q	cerrar el editor
CTRL+Z	cancelar cambios
CTRL+MAJ+Z	de-cancelar cambios
CTRL+I	saltar a una línea dada
MAJ+flechas	seleccionar texto
CTRL+A	seleccionar todo el contenido del archivo
CTRL+C	copiar seleccionado
CTRL+X	cortar
CTRL+V	pegar

Configuración de `gedit`

Activar las opciones siguientes en Editar > Preferencias:

1. mostrar los números de línea
2. insertar espacios en lugar de tabuladores

Esto es porque la sintaxis de Haskell depende de la *indentación* (ie el numero de espacios al principio de cada línea).

Repaso GHCI

Se ejecuta desde el terminal, tipeando: `ghci`.

```
$ ghci
GHCi, version 7.6.3: http://www.haskell.org/ghc/  :? for help
Loading package ghc-prim ... linking ... done.
Loading package integer-gmp ... linking ... done.
Loading package base ... linking ... done.
Prelude>
```

Dentro de `ghci`

Tiene unos atajos parecidos a los del terminal:

CTRL+L limpiar pantalla
CTRL+D salir

Y comandos que empiezan con ":"

:e archivo.hs	editar archivo
:e	editar el mismo
:l archivo.hs	interpretar archivo
:t expresion	ver el tipo de una expresión
:info expresion	más información
:browse modulo	listea contenido de módulo
:?	ver ayuda

Ejercicio

- borrar los archivos creados durante este teórico
- crear una carpeta para esta materia (sin espacios en el nombre, usar _ si quieren o EscribirAsiSinEspacios)
- dentro de esa carpeta, crear una carpeta para el proyecto 1
- dentro de esa carpeta, crear un archivo `proyecto1.hs`
- organisarlo por secciones:

```
-- ejercicio 1

[codigo haskell]

-- ejercicio 2

[etc]
```

Lecturas recomendadas

- Introducción a Linux