

# Introducción a Unix

Sistemas Operativos  
DC - UBA - FCEN

15 de Marzo de 2016

# Prerrequisitos (de la próxima clase)

Supondremos que a esa altura no deberían tener problemas para:

- conectarse por ssh
- moverse por el filesystem
- operar con archivos y dirs.
- distinguir “allá” de “acá”
- editar un archivo de texto
- escribir un `hello.c`
- lograr compilarlo
- lograr ejecutarlo
- guardar salida a archivo
- salida normal vs. errores
- filtrar líneas de texto
- buscar comandos
- buscar syscalls
- buscar ayuda
- buscar `stdlib`
- RTFM

- Intérprete de comandos
- Interfaz de texto.
- Ejemplos: `sh`, `csch`, `ksh`, `bash`.

- Intérprete de comandos
- Interfaz de texto.
- Ejemplos: `sh`, `csch`, `ksh`, `bash`.

Archivos especiales:

- `stdout` Salida standard.

- Intérprete de comandos
- Interfaz de texto.
- Ejemplos: `sh`, `csh`, `ksh`, `bash`.

Archivos especiales:

- `stdout` Salida standard.
- `stderr` Salida standard (errores).

- Intérprete de comandos
- Interfaz de texto.
- Ejemplos: `sh`, `csch`, `ksh`, `bash`.

Archivos especiales:

- `stdout` Salida standard.
- `stderr` Salida standard (errores).
- `stdin` Entrada standard.

- `pwd` Print Working Directory.

- `pwd` Print Working Directory.
- `who` Quién está logeado.



- `pwd` Print Working Directory.
- `who` Quién está logeado.
- `uptime` Cuánto tiempo lleva prendido el sistema.

- `pwd` Print Working Directory.
- `who` Quién está logeado.
- `uptime` Cuánto tiempo lleva prendido el sistema.
- `uname` Qué kernel de Linux esta ejecutando.

- `pwd` Print Working Directory.
- `who` Quién está logeado.
- `uptime` Cuánto tiempo lleva prendido el sistema.
- `uname` Qué kernel de Linux esta ejecutando.
- `man` Muestra las páginas del manual.

- `pwd` Print Working Directory.
- `who` Quién está logeado.
- `uptime` Cuánto tiempo lleva prendido el sistema.
- `uname` Qué kernel de Linux esta ejecutando.
- `man` Muestra las páginas del manual.
- `apropos` Buscador en todas las páginas del manual.

# Manejo de archivos y directorios

- Directorio absoluto: `/home/usuario`.
- Directorio relativo: `../` o `./` o nada.

# Manejo de archivos y directorios

- Directorio absoluto: `/home/usuario`.
- Directorio relativo: `../` o `./` o nada.

Comandos:

- `ls` Lista archivos (dir actual o parámetro).

# Manejo de archivos y directorios

- Directorio absoluto: `/home/usuario`.
- Directorio relativo: `../` o `./` o nada.

## Comandos:

- `ls` Lista archivos (dir actual o parámetro).
- `cd` Cambia el directorio (parámetro o Home).

# Manejo de archivos y directorios

- Directorio absoluto: `/home/usuario`.
- Directorio relativo: `../` o `./` o nada.

## Comandos:

- `ls` Lista archivos (dir actual o parámetro).
- `cd` Cambia el directorio (parámetro o Home).
- `cp`, `mv` Copia/Mueve un archivo o directorio.



# Manejo de archivos y directorios

- Directorio absoluto: `/home/usuario`.
- Directorio relativo: `../` o `./` o nada.

## Comandos:

- `ls` Lista archivos (dir actual o parámetro).
- `cd` Cambia el directorio (parámetro o Home).
- `cp`, `mv` Copia/Mueve un archivo o directorio.
- `rm` Remueve un archivo/directorio.

# Manejo de archivos y directorios

- Directorio absoluto: `/home/usuario`.
- Directorio relativo: `../` o `./` o nada.

## Comandos:

- `ls` Lista archivos (dir actual o parámetro).
- `cd` Cambia el directorio (parámetro o Home).
- `cp`, `mv` Copia/Mueve un archivo o directorio.
- `rm` Remueve un archivo/directorio.
- `mkdir` Crea un directorio.

# Manejo de archivos y directorios

- Directorio absoluto: `/home/usuario`.
- Directorio relativo: `../` o `./` o nada.

## Comandos:

- `ls` Lista archivos (dir actual o parámetro).
- `cd` Cambia el directorio (parámetro o Home).
- `cp`, `mv` Copia/Mueve un archivo o directorio.
- `rm` Remueve un archivo/directorio.
- `mkdir` Crea un directorio.
- `rmdir` Elimina un directorio.

# Manejo de archivos y directorios

- `cat` Muestra por stdout el contenido de un archivo.

# Manejo de archivos y directorios

- `cat` Muestra por stdout el contenido de un archivo.
- `less` Muestra por stdout el contenido de un archivo (paginado).

# Manejo de archivos y directorios

- `cat` Muestra por stdout el contenido de un archivo.
- `less` Muestra por stdout el contenido de un archivo (paginado).
- `echo` Escribe en stdout lo indicado por parámetro.
- `head` Escribe en stdout las primeras líneas de un archivo.
- `tail` Escribe en stdout las últimas líneas de un archivo.

# Busqueda, Matching y Procesamiento

- `find` Busca dentro del arbol de directorio.

```
find /home -name "*.c" -exec ls -al {} \;
```

# Busqueda, Matching y Procesamiento

- **find** Busca dentro del arbol de directorio.

```
find /home -name "*.c" -exec ls -al {} \;
```

- **grep** Busca matching de cadenas en archivos.

```
grep "hola" archivo.txt
```



# Busqueda, Matching y Procesamiento

- **find** Busca dentro del arbol de directorio.

```
find /home -name "*.c" -exec ls -al {} \;
```

- **grep** Busca matching de cadenas en archivos.

```
grep "hola" archivo.txt
```

- **sed** Filtra y transforma texto.

```
sed -i 's/hola/chau/g' archivo.txt
```

# Busqueda, Matching y Procesamiento

- **find** Busca dentro del arbol de directorio.

```
find /home -name "*.c" -exec ls -al {} \;
```

- **grep** Busca matching de cadenas en archivos.

```
grep "hola" archivo.txt
```

- **sed** Filtra y transforma texto.

```
sed -i 's/hola/chau/g' archivo.txt
```

- **awk** Procesamiento de texto.

```
awk '{ print $1 }' archivo.txt
```

```
ls -l | awk '{ print $1 }'
```

```
ls -l | awk '{total += $5} END {print total}'
```

## Redirecciones:

- > Redirige stdout a un archivo.

```
ls > lista_de_archivos.txt
```

## Redirecciones:

- **>** Redirige stdout a un archivo.

```
ls > lista_de_archivos.txt
```

- **&>** Redirige stdout y stderr a un archivo.

```
ls &> lista_de_archivos.txt
```

## Redirecciones:

- **>** Redirige stdout a un archivo.

```
ls > lista_de_archivos.txt
```

- **&>** Redirige stdout y stderr a un archivo.

```
ls &> lista_de_archivos.txt
```

- **>>** Redirige stdout a un archivo en modo **append**.

```
ls >> lista_de_archivos.txt
```

## Redirecciones:

- **>** Redirige stdout a un archivo.

```
ls > lista_de_archivos.txt
```

- **&>** Redirige stdout y stderr a un archivo.

```
ls &> lista_de_archivos.txt
```

- **>>** Redirige stdout a un archivo en modo **append**.

```
ls >> lista_de_archivos.txt
```

- **&>>** Redirige stdout y stderr a un archivo en modo **append**.

```
ls &>> lista_de_archivos.txt
```

## Redirecciones:

- **>** Redirige stdout a un archivo.

```
ls > lista_de_archivos.txt
```

- **&>** Redirige stdout y stderr a un archivo.

```
ls &> lista_de_archivos.txt
```

- **>>** Redirige stdout a un archivo en modo **append**.

```
ls >> lista_de_archivos.txt
```

- **&>>** Redirige stdout y stderr a un archivo en modo **append**.

```
ls &>> lista_de_archivos.txt
```

- **<** Redirige stdin.

```
sort < lista_de_archivos.txt
```

## Redirecciones:

- **>** Redirige stdout a un archivo.  
`ls > lista_de_archivos.txt`
- **&>** Redirige stdout y stderr a un archivo.  
`ls &> lista_de_archivos.txt`
- **>>** Redirige stdout a un archivo en modo **append**.  
`ls >> lista_de_archivos.txt`
- **&>>** Redirige stdout y stderr a un archivo en modo **append**.  
`ls &>> lista_de_archivos.txt`
- **<** Redirige stdin.  
`sort < lista_de_archivos.txt`
- **|** Pipe. Copia stdout a stdin.  
`cat lista_de_archivos.txt | sort`





---

`http://explainshell.com/`

Ejemplo:

```
$ ls -hl
```

```
-rw-rw-r-- 1 user group 445 mar 14 16:12 archivo
```

```
drwxrwxr-x 2 user group 4,0K mar 14 19:31 directorio
```

Ejemplo:

```
$ ls -hl
```

```
-rw-rw-r-- 1 user group 445 mar 14 16:12 archivo
```

```
drwxrwxr-x 2 user group 4,0K mar 14 19:31 directorio
```

Permisos:

- **r** Read
- **w** Write
- **x** eXecute

Entidad:

- **u** User
- **g** Group
- **o** Others

Ejemplo:

```
$ ls -hl
```

```
-rw-rw-r-- 1 user group 445 mar 14 16:12 archivo  
drwxrwxr-x 2 user group 4,0K mar 14 19:31 directorio
```

Permisos:

- **r** Read
- **w** Write
- **x** eXecute

Entidad:

- **u** User
- **g** Group
- **o** Others

Comandos:

- **chown christian:so archivo.txt**
- **chmod u+x archivo.txt**

- `vi/vim` Vi/Vim
- `nano` Nano

- `ssh` Secure Shell
- `scp` Secure Shell Copy
- `sshfs` Secure Shell File System: para montar carpetas remotas.

¿Qué hace la herramienta `nmap`?

¿Qué hace la herramienta `nmap`?

Respuesta:

RTFM



¿Qué hace la herramienta `nmap`?

Respuesta:

RTFM

```
man nmap
```

¿Qué hace la herramienta `nmap`?

Respuesta:

RTFM

```
man nmap
```

¿Qué número ASCII tenía la 'á'?

# Ciclo básico de trabajo en UNIX

¿Qué hace la herramienta `nmap`?

Respuesta:

RTFM

```
man nmap
```

¿Qué número ASCII tenía la 'á'?

Respuesta:

RTFM

# Ciclo básico de trabajo en UNIX

¿Qué hace la herramienta `nmap`?

Respuesta:

RTFM

```
man nmap
```

¿Qué número ASCII tenía la 'á'?

Respuesta:

RTFM

```
man ascii
```

# ¿Preguntas?