

Linuxsurvival.

Módulo uno

Nivel 1

Linux Survival is a free tutorial designed to make it as easy as possible to learn Linux. Even though Linux has hundreds of commands, there are only about a dozen you need to know to perform most basic tasks.

This tutorial uses a simulated Linux terminal so you can practice what you learn. To try it out, type "ls" (without the quotation marks) next to the "zoo>" prompt in the terminal and then hit Enter.

```
zoo> ls  
chimps      cobras      elephants  
gorillas    sq_monks
```

Correct.

OK

animals

Nivel 2

Now, let's see what this would look like in a Linux interface.

The black window is a representation of a Linux command-line interface. The "zoo>" prompt shows you where to type your command.

Type the command which lists the contents of your current directory, then hit the Enter key. Remember to type the command in lower-case.

```
zoo> ls  
chimps      cobras      elephants  
gorillas    sq_monks
```

Correct.

OK

animals



Nivel 3

Okay, it's time to start organizing our animal files. First, we'll create a directory under "animals" called "primates". The command to create a directory is "mkdir" which is short for "make directory".

For example, to make a directory named dogs, you would type

```
mkdir dogs
```

Now type the command to create directory "primates".

```
zoológico> primates mkdir  
zoológico> ls  
chimpancés cobras elefantes  
gorilas primates sq_monks
```

Correcto.

DE ACUERDO

animals



primates

Nivel 4

Now, let's move all of the primate files into our newly created directory.

To move a file, you just use the "mv" command. For example, to move a file called "wolves" into directory "dogs", you would type

```
mv wolves dogs
```

Renaming files is simply a case of "moving" a file from one name to another. For example, to rename file "wolves" to "coyotes", you would type

```
mv wolves coyotes
```

Let's start by moving "chimps" into the "primates" directory. Type the command to move file "chimps" into directory "primates".

```
zoo> mv chimps primates  
zoo> ls  
cobras      elephants      gorillas  
primates    sq_monks
```

Correct.

OK

animals



primates

Nivel 5

Rather than let things get tedious by having you type the commands to move the other two primate files into the "primates" directory, we'll assume that you've already typed them.

Now, we want to go into the "primates" directory and admire our handiwork. To change directories, use the "cd" command, which stands for -- you guessed it -- "change directory".

For example, to change to directory "dogs", you would type

```
cd dogs
```

At the "zoo>" prompt, type the command to change to directory "primates".

```
zoo> cd primates
```



animals



Nivel 6

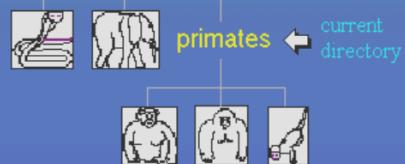
Some people modify their personal Linux configuration so that whenever they change to another directory, the command prompt changes to reflect it. The prompt shown to the right is an example of one. It lists the machine name first (zoo), then a ':', and finally the current directory (primates). If you do not have this sort of configuration, then you will need to learn a command which tells you where you are in the directory structure.

To find out where you are, use the "pwd" command, which stands for "print working directory".

Type it at the command prompt to verify your current location. Then, at the second prompt, type the command which lists the contents of the current directory, so we can be sure that everything was moved into this directory properly.

```
zoo:primates> pwd  
/animals/primates  
zoo:primates> ls  
chimps      gorillas      sq_monks
```

animals



Nivel 7

Type the command to change to directory "animals", then type the command which prints your working directory to make sure you ended up in the right place.

```
zoo:primates> cd ..
zoo> pwd
/animals
```

Good.
OK

```
graph TD
    animals[animals] --> primates[primates]
    animals --> reptiles[reptiles]
```

The diagram shows a folder named 'animals' with a blue arrow pointing to it labeled 'current directory'. Inside 'animals' are two subfolders: 'primates' and 'reptiles'.

Nivel 8

You've already accomplished quite a bit of organizing. Now, you might want to create a place to put your "cobras" file. Suppose you choose to put it in a directory called "reptiles".

You can accomplish this in two steps:

- 1) Make directory "reptiles".
- 2) Move the "cobras" file into directory "reptiles".

Type the commands to execute the two actions listed above; then we'll have a quiz on the commands you've learned so far.

```
zoo> mkdir reptiles
zoo> mv cobras reptiles
```

Excellent.
OK

```
graph TD
    reptiles[reptiles] --> cobras[cobras]
```

The diagram shows a folder named 'reptiles' with a blue arrow pointing to it labeled 'current directory'. Inside 'reptiles' is a file named 'cobras'.

Quiz numero 1

Quiz Number 1

¿Qué comando se utiliza para:

- 1) ¿Cambiar a otro directorio?
- 2) ¿Mover archivos y directorios?
- 3) ¿Cambiar el nombre de archivos y directorios?
- 4) ¿Listar el contenido de un directorio?
- 5) ¿Imprimir su directorio de trabajo actual?
- 6) ¿Crear un directorio nuevo?
- 7) ¿Mostrar el contenido de un archivo?

| | |
|-------|---|
| cd | ✓ |
| mv | ✓ |
| mv | ✓ |
| ls | ✓ |
| pwd | ✓ |
| mkdir | ✓ |
| ls | ✓ |

más

COMPROBAR RESPUESTAS

Módulo dos

Nível 1

Now we have to admit that our suggestions on the page before Quiz Number 1 (in Module 1) were not very good because the category "reptiles" is too broad. It would make more sense to have more specific directories such as "snakes", "lizards", etc. You probably would not have made this mistake.

There is a very simple way to accomplish this task, but we want to demonstrate a few more concepts and commands, so we're going to use a slightly more complicated method, and we'll show you the easy way when we're finished.

First make directory "snakes".

```
zoo> mkdir snakes
```

Good job.

OK

animals



Nível 2

All right, enough theory -- it's time for practice. We need to move "cobras" from the "reptiles" directory to the "snakes" directory, but out of paranoia, we're going to copy it and then remove the original file after we're sure that it was copied correctly. The copy command is "cp" and it has the same syntax as the "mv" command. For example:

```
cp apples apples2
```

We're not in the "reptiles" directory, so you'll have to use a relative pathname in your copy command. Type the command to copy "cobras" from the "reptiles" directory to the "snakes" directory.

```
zoo> cp reptiles/cobras snakes
```

Excellent. This was a tough one.

OK

animals ← current directory



Nivel 3

Let's assume that you've checked the copy of "cobras" which is in the "snakes" directory and everything is okay. We should now remove the original "cobras" file in the "reptiles" directory. The remove command is "rm". For example, to remove file "platypus", you would type

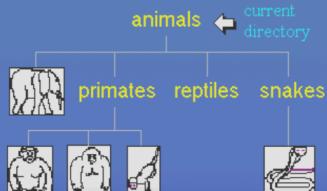
```
rm platypus
```

Type the command to remove the "cobras" file from the "reptiles" directory.

```
zoo> rm reptiles/cobras
```

That's right.

OK



Nivel 4

Now we should remove the "reptiles" directory. The "remove directory" command is "rmdir". For example, to remove directory "fish", you would type

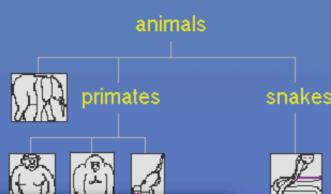
```
rmdir fish
```

Type the command to remove directory "reptiles".

```
zoo> rmdir reptiles
```

Correct.

OK

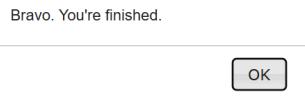


Nivel 5

Okay, now that we've shown you a fairly painful way to accomplish this task, it's time to reveal the easy way. You could have just typed "mv reptiles snakes" because the "mv" command can be used to rename directories as well as files. I hope you're not too upset about being led down the garden path -- it was a useful exercise.

Now we have just one task left and that is to create a directory called "pachyderms" and move "elephants" into it. Type the necessary commands and you'll be finished organizing all of the animal files.

```
zoo> mkdir pachyderms  
zoo> mv elephants pachyderms
```



Nivel 6

Okay, time to get back into action. Type the command to give "write" permission to the prim "group" for file "chimps".

Then type the command to give a long listing of the files so you can check your handiwork.

```
zoo:primates> chmod g+w chimps  
zoo:primates> ls -l  
-rw-rw-r-- keeper prim 547 9:31 chimps  
-rw-rw-r-- keeper prim 983 9:32 gorillas  
-rw-rw-r-- keeper prim 485 9:34 sq_monks
```

| | - - - | - - - | - - - |
|-----------------|-------|-------|------------------|
| | r w x | r w x | r w x |
| user (owner) | | group | other (world) |

Nivel 7

Type the command to give "write" permission to the prim "group" on "all files" in the current directory.

Then type the command to give a long listing of the files.

```
zoo:primates> chmod g+w *  
zoo:primates> ls -l  
-rw-rw-r-- keeper prim 547 9:31 chimps  
-rw-rw-r-- keeper prim 983 9:32 gorillas  
-rw-rw-r-- keeper prim 485 9:34 sq_monks
```

| | - - - | - - - | - - - |
|-----------------|-------|-------|------------------|
| | r w x | r w x | r w x |
| user (owner) | | group | other (world) |

Nivel 8

The default Linux security model is a bit inflexible. To give special access (such as modification privileges) to a group of people, you have to get your system administrator to create a group with those people in it. Furthermore, if you would like to give a different set of access privileges (such as read access) to another group of people, you can't do it because you can only assign one group owner per file or directory. To solve this problem, you can use ACLs (Access Control Lists), a topic which is beyond the scope of this tutorial.

While we're on the subject of groups, we should see which groups you're in. To get a listing of your group memberships, type

groups

Try it at the command prompt.

```
zoo:primates> groups  
admin      all-users
```

Right.

OK

Quiz 2

Which command is used to:

- 1) Copy files?
- 2) Remove files?
- 3) List the groups you are in?
- 4) Remove empty directories?
- 5) Change file permissions?

| | |
|--------|---|
| cp | ✓ |
| rm | ✓ |
| groups | ✓ |
| rmdir | ✓ |
| chmod | ✓ |

- 1) Which wildcard represents all files?
- 2) Which letter represents all other users?
- 3) Which letter represents execute permission?

| | |
|---|---|
| * | ✓ |
| o | ✓ |
| x | ✓ |

CHECK ANSWERS

Page 31



Módulo 3

Nivel 1

If you wanted to figure out which command is used to spell-check a document, you would type

man -k spell

This would show you every command which has the letters "spell" somewhere in its brief description.

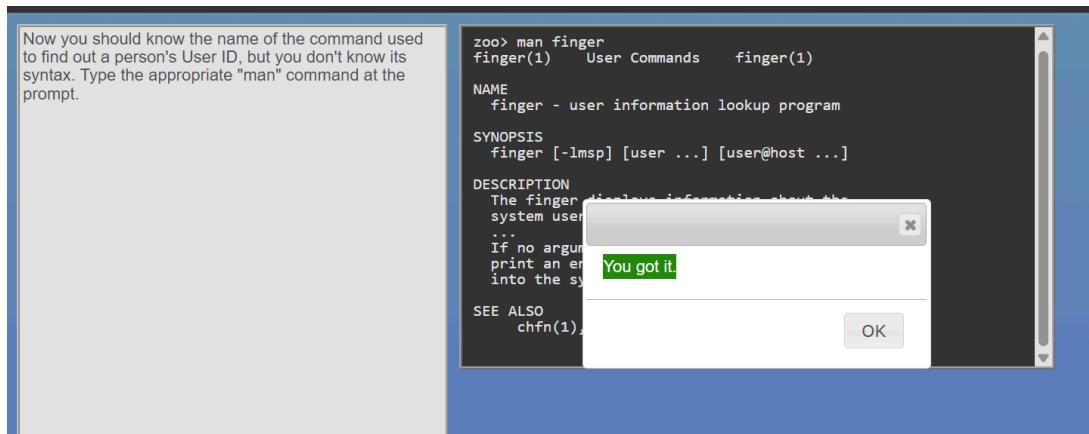
Since you want to find the command which will tell you Greg's User ID, you should probably do a manual search on the word "user". Type the appropriate command at the prompt.

```
zoo> man -k user  
finger (1) - user information lookup program  
newusers (8) - update and create new users  
write (1) - send a message to another user
```

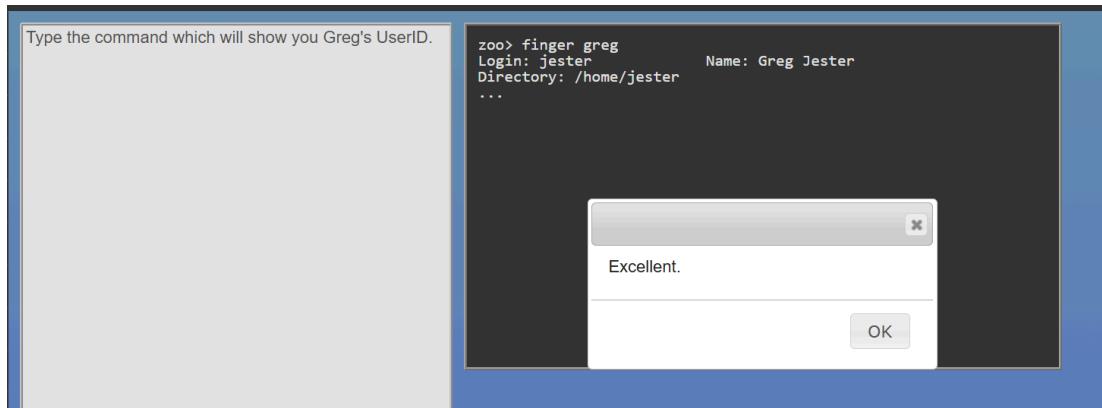
That's it.

OK

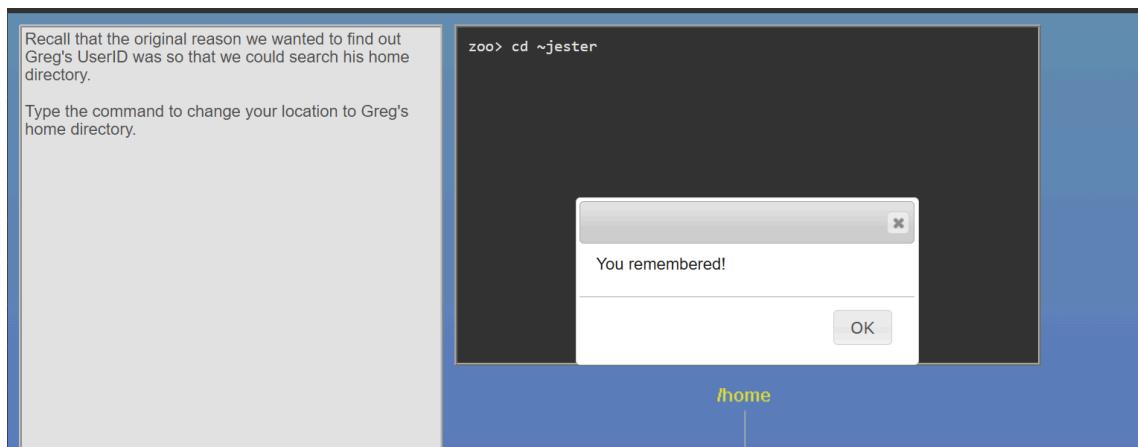
Nivel 2



Nivel 3



Nivel 4



Nivel 5

Here is that example again:
find ~ -name "poem"

Now it's time to try your luck with "find". By the way, you're pretty sure that Greg's joke files begin with the letters "joke".

```
zoo:~jester> find . -name "joke*"  
.kids/joke-1  
.kids/joke-2
```

Great job! That was the most difficult command in this course.

OK



Nivel 6

First, go into the directory where Jester's joke files are stored.

Second, print the contents of "joke-1" and "joke-2" to your screen, using "cat".

```
zoo:~jester> cd kids  
zoo:~jester/kids> cat joke-1 joke-2  
How do you catch a unique rabbit?  
You "neek" up on him.  
  
How do you catch a tame rabbit?  
Tame way.
```

Good.

OK



Nivel 7

Here is the example from the previous page:
cat jabber wocky > poem

At the command prompt, type the command which will concatenate files "joke-1" and "joke-2" and send the output to a file in your home directory called "corny". We know that we do not already have a file named "corny", so don't use the append operator.

```
zoo:~jester/kids> cat joke-1 joke-2 > ~/corny  
zoo:~jester/kids> cd ~
```

Correct.

OK

~/corny

Nivel 8

Our example on the previous page was:

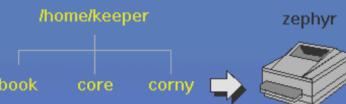
lpr -P hp14 thoughts

Type the command to send file "corny" to printer "zephyr".

```
zoo:~> lpr -P zephyr corny
```

Yes.

OK



Nivel 9

Type the command to show the status of all jobs in the "zephyr" queue.

```
zoo:~> lpq -P zephyr
zephyr is ready and printing
Rank   Owner    Job  Files      Total Size
active  jester   736  pranks    1192 bytes
1st    keeper    737  corny     305 bytes
```

You've got it.

OK



Nivel 10

We will show the results of the previous command on this page, so you don't have to flip back and forth.

Uh oh. You just remembered that "zephyr" is a color plotter. Since it would be very wasteful to use such an expensive device to print out jokes for your children, you decide that you had better cancel your print job before anyone notices. The command to remove print jobs is called "lprm", which stands for "line printer remove". For example, to remove print job 148 from print queue "hp14", you would type

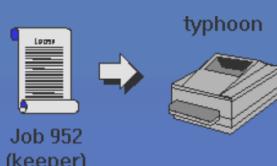
lprm -P hp14 148

Type the command to cancel your print job.

```
zoo:~> lpq -P zephyr
zephyr is ready and printing
Rank   Owner    Job  Files      Total Size
active  jester   736  pranks    1192 bytes
1st    keeper    737  corny     305 bytes
zoo:~> lprm -P zephyr 737
dfa737Aa00853 dequeued
cfa737Aa00853 dequeued
zoo:~> lpr -P typhoon corny
```

It's printing.

OK



Quiz 3

Quiz Number 3

Which command is used to:

- 1) Concatenate files?
- 2) View manual pages?
- 3) Show print queue status?
- 4) Locate files?
- 5) Remove print jobs?
- 6) Show user information?

| | |
|--------|---|
| cat | ✓ |
| man | ✓ |
| lpq | ✓ |
| find | ✓ |
| lprm | ✓ |
| finger | ✓ |

- 1) Which symbol means home directory?
- 2) Which symbol sends output to a new file?
- 3) Which symbol means current directory?

| | |
|---|---|
| ~ | ✓ |
| > | ✓ |
| . | ✓ |

Módulo 4

Nivel 1

Maybe we should give up on Greg Jester for a while and try to get some more interesting files from another friend. It so happens that your friend Belinda Bookie has several sports pools which she keeps online. You decide to copy Belinda's "pools" directory to your home directory, so you can set up your own pools.

The regular "cp" command will not let you copy directories, but if you use the "-r" option, it will. For example, if you wanted to copy a directory called "jokes" from Jester's home directory to your own, you would type

```
cp -r ~jester/jokes ~
```

This would copy the entire "jokes" directory tree.

Type the command which will copy Belinda's "pools" directory to your home directory.

```
zoo:~> cp -r ~bookie/pools ~
cp: No space left on device
```

You're right.

OK

Nivel 2

Type the command which gives a disk usage listing.

```
zoo:~> df
1k-blocks      Used Available Use% Mounted on
131072        62250     62258  50% /
524288        403440    94634  81% /usr
786432        747110         0 100% /home
```

Correct.

OK

Nivel 3

One problem with the "df" command is that you get a list of all of the disks on your system, rather than just the one that you are interested in. Even though it was pretty easy to determine which disk contained your home directory on the previous page, it can actually be quite difficult at times to determine which disk particular files are stored on. Fortunately, there is a way to get the "df" command to figure out which disk contains a particular file or directory and then display statistics only for that disk. For example, if you wanted to get statistics for the disk where your home directory resides, you would type

```
df ~
```

Type the command to find out how much disk space is free on the disk where your current directory resides.

```
zoo:~> df .
1k-blocks      Used Available Use% Mounted on
786432          747110            0 100% /home
```

You're right.

OK

Nivel 4

When your "cp -r" command failed due to a lack of disk space, it did not clean up after itself. That is, it did not delete the files which it managed to copy before running out of space. It would be a good idea to clean up this mess ourselves.

Recall that the "rmdir" command only deletes empty directories, so we cannot use it to delete the "pools" directory in one fell swoop, since "pools" contains at least one file. The command that allows you to remove an entire directory tree is "rm -r". For example, to remove a directory called "stocks" and all of its subdirectories and files, you would type

```
rm -r stocks
```

Type the command to remove your incomplete "pools" directory.

```
zoo:~> rm -r pools
```

Good. It has been removed.

OK



Nivel 5

Type the command to list all processes.

```
zoo:~> ps aux
USER     PID   START   TIME  COMMAND
root      1 15:30  0:04  init
root      2 15:30  0:00  [keventd]
root      3 15:30  0:00  [ksftirqd_CPU0]
...
keeper   741 16:28  0:01  pine
keeper   796 16:53  0:05  rogue
```

Yes.

OK

Nivel 6

The example on the previous page was:

```
cat joke-1 joke-2 | grep rabbit
```

Okay, enough talk. Recall that our goal is to kill a runaway process called "rogue". Type the command sequence which will list only those processes which contain the word "rogue".

```
zoo:~> ps aux | grep rogue
keeper  796 16:53  0:05 rogue
keeper  825 17:04  0:00 grep rogue
```

Excellent. That was a difficult one.

OK

Nivel 7

Now all we have to do is kill "rogue" before it craters our Linux system. First note the PID of the "rogue" process. This is your cue to say, "Hey, what happened to the column headings?" Well, "grep" didn't display the line of headings because it didn't contain the word "rogue". Luckily, it's easy to remember which column contains the PID -- it's the first one with a number (i.e. it's the column to the right of "keeper").

To kill a process, simply type

```
kill PID
```

where PID is the ID of the process you want to kill.

Type the command to kill the "rogue" process.

```
zoo:~> ps aux | grep rogue
keeper  796 16:53  0:05 rogue
keeper  825 17:04  0:00 grep rogue
zoo:~> kill 796
```

Right.

OK

Nivel 8

To see whether "rogue" actually died or not, we need to look for the "rogue" command in a list of processes again.

Execute the command to see whether or not the "rogue" process is still running.

```
zoológico:~> ps aux | grep rogue
guardián 796 16:53 0:06 picaro
guardián 889 17:05 0:00 grep rogue
```

Buen trabajo.

DE ACUERDO

Nivel 9

When you use "kill" on a process, "kill" tries to shutdown the process gracefully. That is, it allows the process to clean up after itself before it goes away. However, sometimes (as was the case with "rogue") the process is so messed up that it ignores the signal to die. That's why there is an option to the "kill" command which means "kill immediately". To tell a process to die immediately, type

```
kill -9 PID
```

where PID is the ID of the process you want to kill.

Type the command to kill "rogue" immediately.

```
zoo:~> ps aux | grep rogue
keeper  796 16:53  0:06 rogue
keeper  889 17:05  0:00 grep rogue
zoo:~> kill -9 796
```

You got it.

OK

Nivel 10

Now we just have to make sure that "kill -9" worked. Believe it or not, sometimes even a "kill -9" won't eliminate a process. If that happens, then it's probably time to give your friendly system administrator a call and let her worry about it.

Type the command to see whether or not the "rogue" process is still running.

```
zoo:~> ps aux | grep rogue  
keeper 898 17:06 0:00 grep rogue
```

Correct.

OK

Quiz 4

Quiz Number 4

Which:

- 1) command shows free disk space?
- 2) command shows process status?
- 3) option to the above command shows all system processes?
- 4) command tells a process to die gracefully?
- 5) option to the above command tells a process to die immediately?
- 6) command finds words in text?
- 7) symbol sends output to another program?
- 8) 'cp' and 'rm' option acts on trees?

| | |
|------|---|
| df | ✓ |
| ps | ✓ |
| aux | ✓ |
| kill | ✓ |
| -9 | ✓ |
| grep | ✓ |
| | ✓ |
| -r | ✓ |

Nivel 1: Se aprende a usar el comando `ls` para listar el contenido del directorio actual.

Nivel 2: Se enfoca en cómo distinguir visualmente archivos de directorios al usar el comando `ls`.

Nivel 3: Se utiliza el comando `pwd` para imprimir el directorio de trabajo actual.

Nivel 4: El comando `cd` es introducido para permitir al usuario cambiar de un directorio a otro.

Nivel 5: Se practican los nombres de ruta relativos, incluyendo el uso de `cd ..` para subir un nivel en la estructura de directorios.

Nivel 6: Se explican los nombres de ruta absolutos, que son aquellos que comienzan con `/` (la raíz del sistema de archivos).

Nivel 7: El foco es la exploración y comprensión del directorio raíz (`/`).

Nivel 8: Se introduce el directorio de inicio (`~`), un atajo útil para volver al directorio *home* del usuario.

Quiz 1: Una evaluación que cubre los comandos `ls`, `cd`, `pwd`, y el uso de rutas relativas y absolutas, `~`, y `/`.

Nivel 1: El comando `touch` es el foco principal, utilizado para crear archivos vacíos.

Nivel 2: Se aprende a crear directorios usando el comando `mkdir`.

Nivel 3: Se introduce el comando `cp` para copiar archivos de un lugar a otro.

Nivel 4: Se utiliza la opción recursiva con `cp -r` para copiar directorios completos.

Nivel 5: El comando `mv` se utiliza para mover archivos y también para renombrarlos.

Nivel 6: Similarmente, se usa `mv` para mover y renombrar directorios.

Nivel 7: Se introduce el comando `rm` para eliminar archivos.

Nivel 8: Se explican dos comandos para la eliminación de directorios: `rmdir` (para directorios vacíos) y `rm -r` (para directorios no vacíos).

Quiz 2: Evaluación de los comandos de manipulación: `touch`, `mkdir`, `cp`, `mv`, `rm`, y `rm -r`.

Nivel 1: Se utiliza el comando `cat` para imprimir todo el contenido de un archivo directamente en la terminal.

Nivel 2: El comando `less` se introduce para ver el contenido de archivos grandes de forma paginada.

Nivel 3: Se usa `head` para mostrar solo las primeras líneas de un archivo.

Nivel 4: El comando `tail` muestra las últimas líneas de un archivo.

Nivel 5: Se aprende a usar `grep` para buscar patrones o cadenas de texto específicas dentro de los archivos.

Nivel 6: Se utiliza el operador de redirección `>` para enviar la salida de un comando a un archivo, sobrescribiendo el contenido existente.

Nivel 7: El operador `>>` se introduce para agregar la salida de un comando al final de un archivo.

Nivel 8: El operador `<` se usa para tomar el contenido de un archivo y pasarlo como entrada a un comando.

Nivel 9: Se introduce el concepto de Tuberías (`|`) para conectar la salida de un comando con la entrada de otro.

Nivel 10: Se practican combinaciones de tuberías con comandos como `grep` para un filtrado de datos más complejo.

Quiz 3: Evaluación de comandos de visualización y redirección: `cat`, `less`, `head`, `tail`, `grep`, y los operadores de redirección (`>`, `>>`, `<` y `|`).

Nivel 1: Introduce el editor de texto **Vi/Vim** y la navegación básica en su Modo Comando.

Nivel 2: Se aprende a abrir, guardar (`:w`), y salir (`:q`, `:wq`) de archivos dentro de Vi/Vim.

Nivel 3: El enfoque se mueve al Modo Inserción usando comandos como `i` (insertar) y `a` (agregar) para la edición de texto.

Nivel 4: Se visualizan y entienden los permisos básicos de archivos (Lectura, Escritura, Ejecución - `r, w, x`) usando `ls -l`.

Nivel 5: Se explican los grupos de permisos: Propietario (`u`), Grupo (`g`), Otros (`o`) y Todos (`a`).

Nivel 6: Se utiliza el método simbólico de `chmod` para cambiar permisos, como `chmod u+x` (agregar permiso de ejecución al propietario).

Nivel 7: El método octal (numérico) de `chmod` se introduce para configuraciones de permisos como `chmod 755`.

Nivel 8: El comando `chown` se enseña para cambiar el propietario de un archivo.

Nivel 9: Se utiliza `chgrp` para cambiar el grupo asociado a un archivo.

Nivel 10: El último nivel repasa cómo eliminar archivos de manera segura y la prevención de sobrescritura.

Quiz 4: Evaluación final que incluye `vi/Vim`, `chmod` (simbólico y octal), `chown`, `chgrp`, y la gestión de permisos.