

Taller práctico de Linux

DESDE EL MANEJO DE COMANDOS HASTA EL DESPLIEGUE
DE AMBIENTES DE PROGRAMACIÓN

Blanca Vázquez

Grupo ArBio

<https://iimas.unam.mx/arblio>

12 de septiembre de 2024

Objetivo general del taller

Aprender comandos esenciales para el manejo de la terminal Linux e instalar ambientes de desarrollo de programación.

Objetivos específicos

- Realizar operaciones entre directorios y archivos en Linux.
- Aprender a usar un editor de línea de comandos.
- Clonar un repositorio.
- Desplegar un ambiente conda usando un archivo .yaml
- Entrenar modelos usando un multiplexor.
- Interactuar con un cuaderno de jupyter desde el servidor.

Acceder a un servidor

1. Abre una terminal.
2. Ejecuta el siguiente comando:

```
ssh username@132.247.19.17
```

3. Escribe la contraseña del usuario asignada.

Operaciones sobre directorios y archivos

Dentro de una terminal, los comandos básicos son:

Comando	Descripción	Ejemplo
pwd	Muestra la ruta de trabajo actual	pwd
ls	Muestra el contenido de un directorio	ls
cd	Moverse entre directorios	cd /oldpath/ /newpath/
mkdir	Crea un nuevo directorio	mkdir newdir
rmdir	Elimina un directorio específico	rmdir mydir
cp	Copia archivos o carpetas	cp sourcefile copyfile
mv	Moverse o renombrar archivos	mv oldName newName

Copiar archivos de local al servidor

Una de las operaciones más comunes es copiar archivos entre una máquina local y un servidor. Copiaremos el archivo holamundo.txt.

Abrir una terminal en el servidor

1. Crea la carpeta: 'servidor'
2. Ubícate dentro de la carpeta y escribe: `pwd`
3. Copia la ruta obtenida.

Abrir una terminal local

1. Clona el repositorio indicado e ingresa a la carpeta: 'local'
2. Ubícate dentro de la carpeta y escribe:
`scp holamundo.txt username@132.247.19.17:/ruta`

En la terminal del servidor, valida el archivo escribiendo: `ls`

Copiar archivos del servidor a local

Abrir una terminal en el servidor

1. Cambia el nombre del archivo de 'holamundo.txt' a 'holamerida.txt' escribiendo: `mv holamundo.txt holamerida.txt`
2. Ubica la ruta del archivo escribiendo: `pwd`
3. Copia la ruta obtenida.

Abrir una terminal local

1. Ubícate en la ruta del archivo a copiar y escribe:
`scp username@132.247.19.17:/ruta/holaiimas.txt .`

En la terminal local, valida el archivo copiado escribiendo: `ls`

Copiar carpetas de local al servidor

Otra de las operaciones comunes es copiar carpetas con múltiples archivos entre una máquina local y un servidor. Vamos a copiar la carpeta: local/

Abrir una terminal en el servidor

1. Ubícate dentro de la carpeta 'servidor/' y escribe: `pwd`
2. Copia la ruta obtenida.

Abrir una terminal local

1. Ubícate en la ruta de la carpeta 'local/' y escribe:
`scp -r local/ username@132.247.19.17:/ruta`

En la terminal del servidor, valida la carpeta copiada escribiendo: `ls`

Copiar carpetas del servidor a local

Abrir una terminal en el servidor

1. Cambia el nombre de la carpeta 'local/' a 'server/' escribiendo:
`mv local/ server/`
2. Ubícate dentro de la carpeta y escribe: `pwd`
3. Copia la ruta obtenida.

Abrir una terminal local

1. Ubícate fuera de la carpeta 'local' y escribe:
`scp -r username@132.247.19.17:/ruta/server/ .`

En la terminal local, valida la carpeta copiada escribiendo: `ls`

Editores de línea de comandos

Existen diferentes editores de línea de comandos en Linux y Unix, los más comunes son: Vim, Emacs y Nano. En particular, Nano¹ es un editor simple y los comandos más usados son:

Ejemplo	Descripción
nano file.txt	Abre un archivo
nano -c file.txt	Abre un archivo y muestra la posición del curso
ctrl + w	Búsqueda de una cadena de texto
ctrl + o	Guarda el archivo
ctrl + k	Corta una línea completa
Esc + 6	Copia una cadena de texto
Esc + U	Semejante a ctrl + z (undo)
Esc + E	Semejante a ctrl + y (redo)

¹Documentación oficial: <https://www.nano-editor.org/docs.php>

El comando *grep*² se utiliza para buscar patrones de texto.

Ejemplo	Descripción
<code>grep "word" file.txt — wc -l</code>	Cuenta las veces que aparece una palabra
<code>grep -i "word" file.txt</code>	Busca una palabra ignorando mayúsculas y minúsculas
<code>grep -v "word" file.txt</code>	Regresa todo el texto, menos la palabra indicada.
<code>grep -n "word" file.txt</code>	Indica el número de línea en donde aparece la palabra

²Documentación oficial:

<https://www.gnu.org/software/grep/manual/grep.html>

- Es una suite de código abierto.
- Multiplataforma.
- Permite utilizar Python y R en un mismo entorno.
- Cuenta con una IDE para desarrollar aplicaciones científicas.
- Aproximadamente cuenta con +1500 paquetes pre-instalados.
- **Conda** es un gestor de paquetes. Permite listar, instalar y desinstalar paquetes instalados en un entorno de Python.
- Conda es el gestor de paquetes por defecto de Anaconda.

Crear un ambiente de programación

Dentro de la carpeta 'conda', ubica el archivo server.yml y cópialo al servidor.

Abrir una terminal en el servidor

1. Después de copiar el archivo, ejecuta el siguiente comando:

```
conda env create --file server.yml
```

2. Para activar el ambiente, ejecuta:

```
conda activate taller-server
```

Entrenando un modelo en el servidor

Copia la carpeta 'scripts/' al servidor dentro de la carpeta 'taller'.

1. Dentro de scripts, abre el archivo 'config.sh' y modifica el número de épocas de entrenamiento.

```
python mnist.py --epochs 5 \  
--dropout 0.2 \  
--layers_dense 10 \  
--optimizer 'adam' \  
--filename 'exp1_adam'
```

2. Guarda los cambios con `ctrl + o` y sal del archivo `ctrl + x`.
3. Gestiona los permisos de ejecución escribiendo:

```
chmod 777 config.sh
```

Iniciar el entrenamiento en un multiplexor

TMUX³ es un multiplexor de terminal que permite realizar tareas en paralelo, entre otras tareas.

1. En tu terminal, escribe: `tmux`
2. Activa el ambiente de conda y muévete a la carpeta de 'taller/scripts'.
3. Ejecuta el archivo bash:
`./config`
4. Renombra el panel con el comando `ctrl + b + ,`
5. Para salir de tmux ejecuta `ctrl + b + d`

³Documentación oficial:

<https://tmuxguide.readthedocs.io/en/latest/tmux/tmux.html>

Al salir de tmux, volverás a la terminal del servidor. Inicia otra sesión de tmux y realiza los siguientes cambios.

1. Abre el archivo 'config.sh' modifica nuevamente el número de épocas, cambia el optimizador de 'adam' por 'sgd' y modifica el valor del parámetro 'filename'.
2. Guarda los cambios y ejecuta el archivo bash.
3. Sal de tmux `ctrl + b + d`

Comandos básicos en tmux

- Para listar las sesiones activas en tmux: `tmux list-sessions`
- Para ingresar a una sesión: `tmux attach -tNÚMERO`
- Cambiar el nombre del panel: `ctrl + b + ,`
- Salir de una sesión (todo se queda en ejecución):
`ctrl + b + d`
- Borrar una sesión (matar / cerrar todos los procesos)
`ctrl + b + x`

TensorBoard⁴ es una suite de herramientas que permiten visualizar y monitorear el rendimiento de los modelos entrenados con Tensorflow.

- Monitoreo de métricas.
- Visualización del grafo del modelo (operaciones y capas).
- Ver histogramas de pesos, sesgos y otros tensores a medida que cambian con el tiempo

⁴Documentación oficial: <https://github.com/tensorflow/tensorboard>

Visualizando el rendimiento de los modelos de manera local

Cerramos todas las terminales y abrimos una nueva terminal:

1. En la nueva terminal escribe:

```
ssh -L 16006:127.0.0.1:6006 username@132.247.19.17
```

2. Usando tmux, abre la sesión del primer modelo que entrenaste:

```
tmux attach -tNÚMERO
```

3. Dentro de tmux, escribe:

```
tensorboard --logdir logs/fit/FILENAME --port=6006
```

4. **NO** cerrar la terminal.

En tu máquina local, abre un explorador (puede ser Chrome o Firefox)

1. En una pestaña del explorador, escribe:

`http://localhost:16006/`

Ejecutando jupyter notebook en el servidor

Ejecutar un cuaderno de jupyter en el servidor e interactuar de forma local.

Abre una terminal en local:

1. Copia la carpeta 'notebooks' hacia el servidor dentro de 'taller'.
2. Cierra la terminal.

Abre una terminal en el servidor:

1. Conéctate al servidor.
2. Abre una nueva sesión en tmux y ubícate dentro de 'notebooks'.
3. Gestiona los permisos para el archivo 'run_jupyter'
`chmod +x run_jupyter.sh`
4. Ejecuta el archivo bash: `./run_jupyter.sh`

Ejecutando jupyter notebook en el servidor

Ejecutar un cuaderno de jupyter en el servidor e interactuar de forma local.

Abre una terminal en local:

1. Creamos un túnel para ejecutar el cuaderno:

```
ssh -N -f -L localhost:8888:localhost:8888  
username@132.247.19.17
```

2. Abrir un explorador (puede ser Chrome) y escribe:

```
http://localhost:8888
```

1) Clonar un repositorio en el servidor.

1. Abrir una terminal, ubicarse en una ruta deseada y ejecutar:

```
git clone https://github.com/blancavazquez/taller-linux.git
```

- Es una plataforma de hospedaje de código para el control de versiones y de colaboración.
- Almacenamiento y administración en nube.
- Control de versiones.
- Amplia comunidad de desarrollo (bien documentado).
- Actúa como una red social para desarrolladores.

1) Clonar un repositorio en el servidor.

1. Abrir una terminal, ubicarse en una ruta deseada y ejecutar:

```
git clone https://github.com/blancavazquez/taller-linux.git
```

Gracias

Blanca Vázquez

blanca.vazquez@iimas.unam.mx

ArBio Group

<https://iimas.unam.mx/arbio>