

SISTEMAS OPERATIVOS

GUÍA PRÁCTICA DE LABORATORIO N°3

GNU-LINUX



Consignas:

1. Inicie sesión con el comando **ssh** en el servidor LINUX cuya dirección IP y hostname están publicados en la cátedra virtual.

```
$ ssh usuario@hostname
```

2. Cree una carpeta llamada “guia3”. e ingrese a la misma

```
$mkdir guia3 ; cd guia3
```

___REDES___

3. El comando “ip”. Este comando, combinado con distintos parámetros, permite conocer bastantes aspectos de la configuración de red, por ejemplo, la IP de un equipo, puertas de enlace, rutas, etc... Analicemos algunos de ellos... ¿que vemos con estos comandos?

3.1. con “ip a” permite modificar y ver información respecto a las placas de red del equipo

```
$ ip a
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
```

```
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
```

```
inet 127.0.0.1/8 scope host lo
```

```
valid_lft forever preferred_lft forever
```

```
inet6 ::1/128 scope host
```

```
valid_lft forever preferred_lft forever
```

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
```

```
link/ether ba:c1:66:e2:58:1f brd ff:ff:ff:ff:ff:ff
```

```
inet 192.168.1.122/20 brd 192.168.15.255 scope global eth0
```

```
valid_lft forever preferred_lft forever
```



```
inet6 fe80::b8c1:66ff:fee2:581f/64 scope link
    valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen
1000
    link/ether 46:95:87:66:8b:a5 brd ff:ff:ff:ff:ff:ff
4: ztygiwyivf: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 2800 qdisc pfifo_fast state
UNKNOWN group default qlen 500
    link/ether 3a:7a:60:f8:11:0e brd ff:ff:ff:ff:ff:ff
    inet 10.100.200.11/24 brd 10.100.200.255 scope global ztygiwyivf
        valid_lft forever preferred_lft forever
    inet6 fe80::387a:60ff:fef8:110e/64 scope link
        valid_lft forever preferred_lft forever
```

3.2. “ip r” permite ver y modificar las rutas

```
$ ip r
default via 192.168.1.1 dev eth0
10.100.200.0/24 dev ztygiwyivf proto kernel scope link src 10.100.200.11
192.168.0.0/20 dev eth0 proto kernel scope link src 192.168.1.122
```

3.3. Archivo interfaces ¿qué observaremos?

```
$cat /etc/network/interfaces
```

3.4. Archivo resolv.conf ¿qué observaremos?

```
$cat /etc/resolv.conf
```

__PROCESOS__

4.1. ps nos permite ver los procesos en ejecución. Para ver todos los procesos en nuestro shell activo ejecutaremos:

```
$ps
PID TTY      TIME CMD
10235 pts/0    00:00:00 bash
10375 pts/0    00:00:00 ps
```

4.2. Para ver todos los procesos del sistema en formato completo ejecutamos:

```
$ps -ef
```

4.3. Integremos conocimientos adquiridos de búsqueda, pipes y comodines... buscando los procesos de nuestro usuario en el listado de procesos del sistema y enviándolo a un archivo llamado “misprocesos.txt”:



FACULTAD DE INGENIERIA

```
$ps -ef | grep relg* > misprocesos.txt
```

4.4. Para ver el consumo de recursos de cada proceso podemos ejecutar:

```
$ps aux
```

4.5. Ver los procesos y sus hilos:

```
$ps -elf
```

4.6. Algo interesante de este comando es que puede personalizarse los atributos que se muestran mediante la opción “axo”. veamos algunos ejemplos:

```
$ps axo user,stat,pid,pcpu,comm
```

```
$ps axo user,pid,ppid,comm
```

4.7. “top” muestra en tiempo real el estado de los procesos del sistema.

```
$top
```

4.8. Para detener un proceso utilizamos el comando “kill”. Existen varias formas de “eliminar” un proceso...

```
$kill -l
```

4.9. Generalmente utilizamos “Kill Signal” por lo tanto vamos a generar un proceso... y a matarlo.

```
$vi hola_<nombre de usuario>
```

la creación de este archivo generará un proceso...
(salir del editor presionando ctrl+z)

```
$ps aux | grep hola_<nombre de usuario>
```

```
$kill -9 PID
```

```
$ps aux | grep hola_<nombre de usuario>
```

```
__DISCOS__
```

5.1. Veamos cuántos discos y particiones tenemos:

```
$lsblk
```



FACULTAD DE INGENIERIA

5.2. ¿Qué sucede si agregamos el parámetro “fm”?

```
$lsblk -fm
```

5.3. Analicemos el comando “df”

```
$df
```

```
$df -hT
```

```
$df -a
```

5.4. Crearemos una carpeta llamada “pruebas” y dentro de ella, agregaremos algunos archivos:

```
$mkdir pruebas ; cd pruebas ; touch cpu disco sistema ram
```

5.5. Agregaremos información al archivo “cpu”

```
$cat /proc/cpuinfo > cpu
```

5.6. Agregamos información al archivo “disco”

```
$df -hT > disco ; lsblk -fm >> disco
```

5.7. Agregamos información al archivo “sistema”

```
$cat /etc/issue.net > sistema ; uname -r >> sistema ; echo _____ >> sistema ; cat /etc/*-release >> sistema
```

5.8. Agregamos información al archivo “ram”

```
$free -h > ram ; vmstat -s -S M >> ram ; echo ..... >> ram ; cat /proc/meminfo >> ram
```

__HERRAMIENTAS EXTERNAS__

Htop es una alternativa a “top” más interactiva y funcional.

```
$htop
```

Glances es una herramienta de monitorización para sistemas operativos Linux que te permitirá tener bajo control los principales aspectos del sistema. Con Glances podremos ver la máxima información en el mínimo espacio posible en consola, además Glances se puede adaptar dinámicamente al tamaño del terminal, de esta forma podremos tener más



FACULTAD DE INGENIERIA

información si ampliamos la ventana. Otra característica muy interesante es que tiene un modo cliente-servidor para no tener que hacerlo presencialmente o a través del protocolo SSH

\$glances