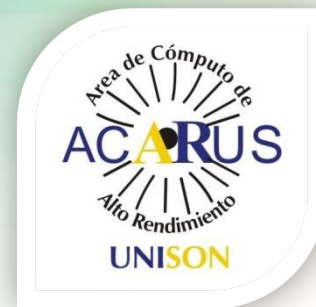


ACARUS

CAR



Responsable :

María del Carmen Heras Sánchez

Asesores Técnicos :

Daniel Mendoza Camacho

Yessica Vidal Quintanar

Día 2:

- ✓ **Conceptos**
- ✓ **Comandos básicos de Linux**
- ✓ **Variables de ambiente**
- ✓ **Módulos**

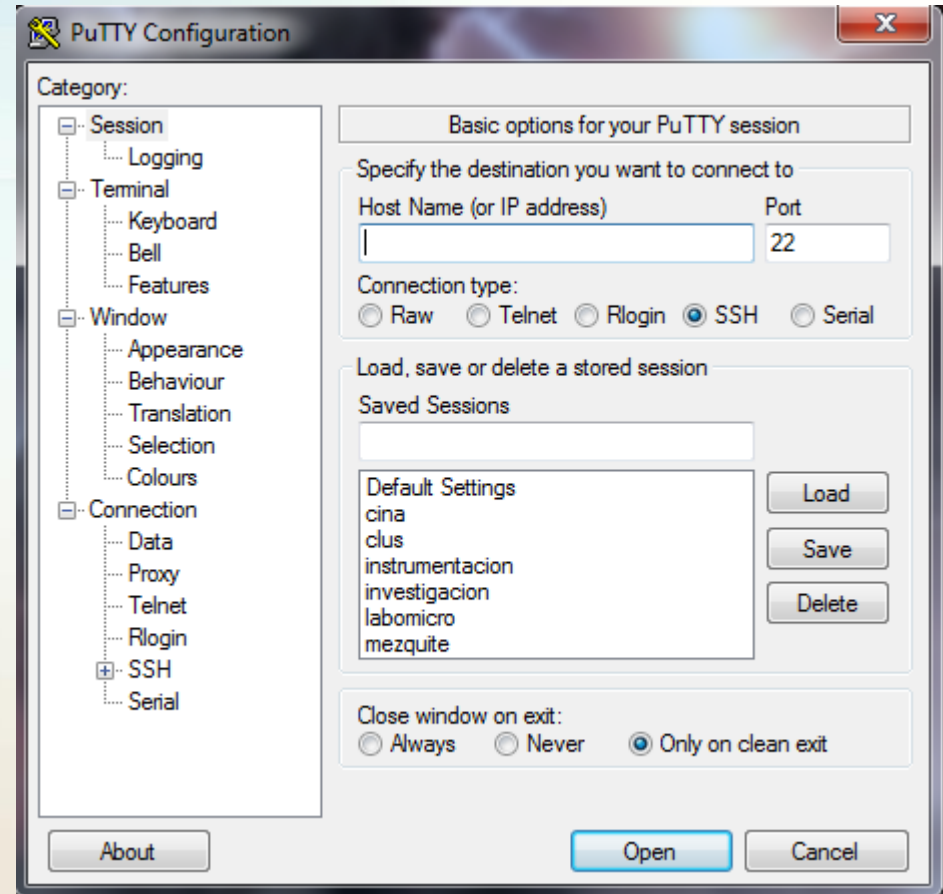
Conexiones

Equipo	Nombre del Cluster	IP
Choya	choya.acarus.uson.mx	148.225.91.79
Ocotillo	ocotillo.acarus.uson.mx	148.225.91.150

Conexión remota

Putty herramienta que permite hacer conexiones via ssh con licencia libre a un servidor. Disponible para varias plataformas

Nota: Es igual para todos los equipos (choya y ocotillo)



Descargar :

<http://acarus.uson.mx/descargas.htm>

Conceptos Básicos

Sistema operativo Linux

Terminal

Interprete de comandos

- Bourne Shell (sh)
- Korn Shell (ksh)
- Bash

Entrada, Salida y Error
Estandar (Flujo de
información)

Estructura de directorios

Rutas absolutas y
relativas



Linux



- Linux es un Sistema Operativo creado por *Linus Torvalds*
- Este sistema lo diseñaron y desarrollaron cientos de programadores , pretendiendo realizar una replica de UNIX.
- Multiplataforma
- Multiusuario y multitarea
- Gratuito (existen distribuciones que se venden Linux Redhat)



Richard Stallman GNU (General No UNIX) **copyleft** -'copia permitida'- (en clara oposición a **copyright** - 'derecho de copia'.

Linus Torvalds empezó a escribir el **núcleo Linux** y decidió distribuirlo bajo la licencia **GPL** General Public License (GPL, Licencia Pública General). el núcleo Linux fue combinado con el sistema GNU, resultando en un sistema operativo libre y completamente funcional. El Sistema Operativo formado por esta combinación es usualmente conocido como "**GNU/Linux**" o como una "**distribución Linux**" y existen diversas variantes.

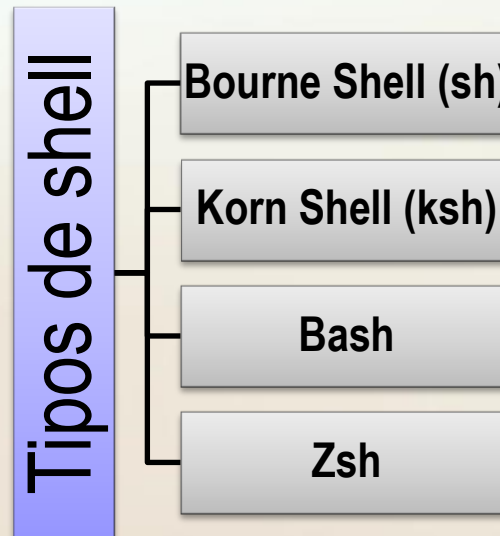
¿Qué es una terminal en Linux?

Linux dispone de un intérprete de órdenes o terminal (en inglés se utiliza la palabra shell) que hace de interfaz entre el usuario y el propio sistema operativo.

Una terminal es una forma de acceder al sistema sin utilizar la interfaz gráfica, es decir, realizar todo tipo de tareas en formato texto. La forma de utilizar el sistema de este modo es mediante órdenes.

Interprete de comandos (shell de texto)

El interprete de comandos actúa como una interfaz alfanumérica y a su vez es un programa informático, hace posible que el sistema operativo realice ordenes deseadas por el usuario mediante una serie de comandos.



Entrada, Salida y Error Estándar (Flujo de información)

- Entrada y salidas, redirecciones y pipes son de las más poderosas herramientas tipo command-line en linux.
- Linux trata las entradas y salidas como cadenas de texto para una fácil manipulación de datos.
- Normalmente la entrada de datos es vía teclado y la salida vía monitor.
- Es posible redireccionar las salidas hacia otros programas como entrada.
- Algo similar se puede hacer con los pipes para mandar la salida de un programa a otro.

Tipos de entrada y salida

- Entrada estándar: Los programas aceptan datos vía teclado o stdin, standard input
- Salida estándar: Los programas en modo texto envían datos de salida a los usuarios vía “standard output”, aka stdout
- Error estándar: Linux provee de un segundo tipo de salida conocida como “standard error” o “stderr”. La idea de esto es regresar información de alta prioridad como lo son mensajes de error.

Redireccionamiento

- Para redireccionar la entrada o salida utilizamos símbolos seguidos del comando incluyendo las opciones que requiera.
- Ejemplo, redireccionamos “echo”
 - `echo $HOSTNAME > salida.txt`
- El resultante es un archivo llamado salida.txt conteniendo el nombre del host del equipo.

Operadores de redirección

Salidas estándar resultados de comandos, de búsquedas, errores

- **>** Crea un archivo nuevo conteniendo la salida estándar, si el archivo existe, lo sobrescribe
- **>>** Agrega la salida estándar a un archivo existente, si el archivo no existe, lo crea
- **2>** Crea un nuevo archivo conteniendo el error estándar, si el archivo existe, lo sobrescribe
- **2>>** Agrega el error estándar a un archivo existente, si el archivo no existe, lo crea
- **&>** Crea un nuevo archivo conteniendo el error y la salida estándar, si el archivo existe, lo sobrescribe

Ejemplos: `cat > file1.txt` `cat >> file1.txt`

Operadores de redirección

- < Manda el contenido de un archivo específico para ser usado como entrada
- << Acepta texto como entrada estándar

Entrada estándar

No es necesario indicar con > o >> porque por default lo hace

Redireccionamiento - Ejemplos

```
> date > fecha.txt  
cat fecha.txt
```

```
>> whoami >> fecha.txt  
cat fecha.txt
```

```
2> ls /root 2> error.txt  
cat error.txt
```

```
2>> ifconfig 2>> error.txt  
cat error.txt
```

```
&> find /home/ -name "*.txt" &> salida.txt
```

```
< tac < archivo.txt
```

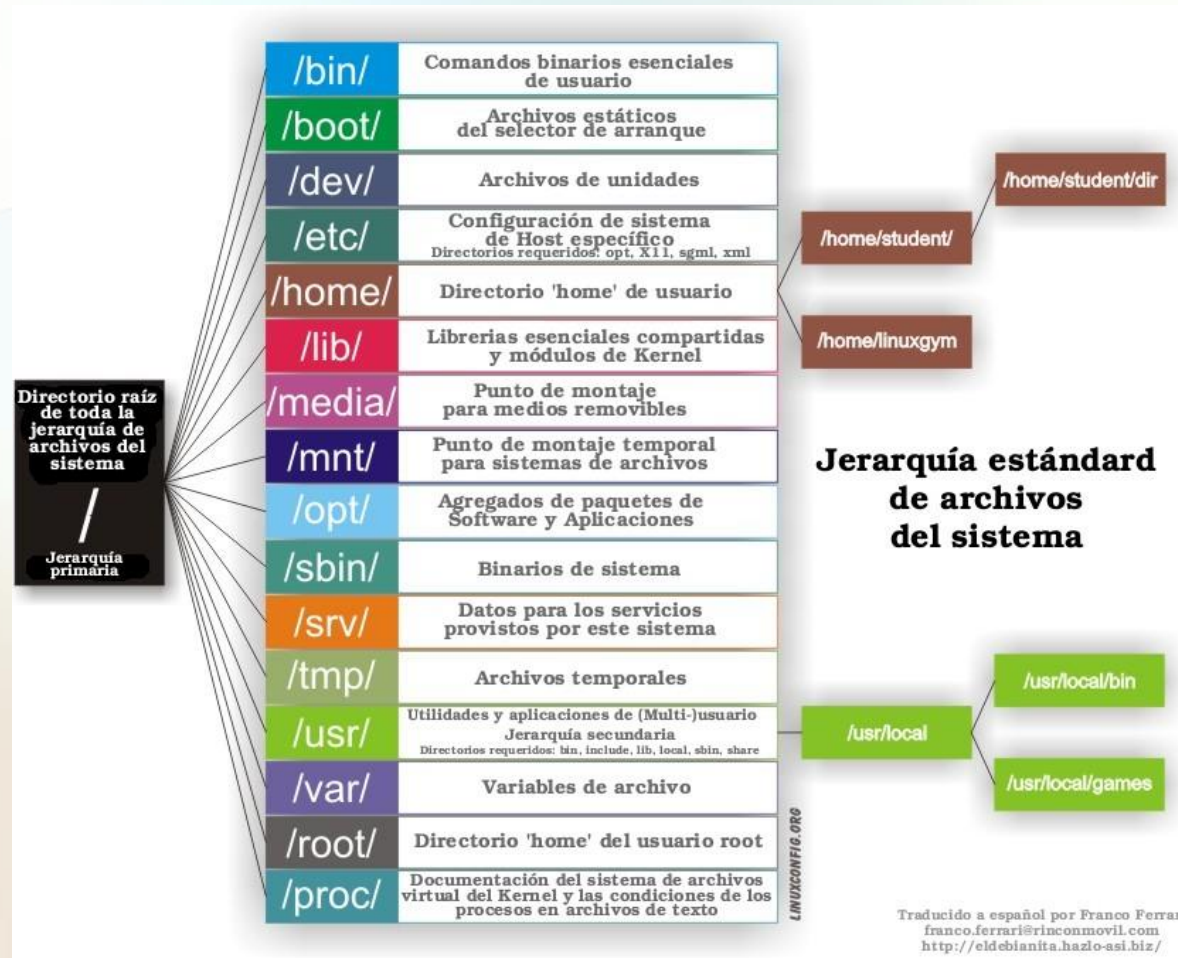
Pipelines o interconexiones

- Algunos programas operan con las salidas de otros programas como los filtros de texto o cualquier otro que manipule la salida de otro.
- Existe una solución llamada pipeline, que redirecciona la salida estándar al segundo programa

Pipelines

- La sintaxis es: `first | second`
- Pueden ser usados en secuencias de comandos muy largos
 - `first | second | third | fourth | fifth | ...`

Estructura de directorios



Trayectorias

- Es la secuencia de directorios que se deben de recorrer para acceder a un archivo o directorio
- Existen dos tipos de trayectorias: absoluta y relativa
- Cada directorio contiene dos directorios especiales, el directorio actual “.” y el directorio padre “..”

Absoluta

- `cd /home/yessica/1/2/3/4/5/` toda la ruta

Relativa

- Partiendo del directorio 3 y se quiere llegar al 6
- `cd ./4/5/6/` o `cd 4/5/6`

Comandos básicos linux

- passwd
- man
- apropos
- ls
- cd
- pwd
- mkdir
- rm
- rmdir
- touch
- Redireccionamiento de la salida estándar
- cat
- head y tail
- more

Comandos básicos de linux

- cp y scp
- mv
- find
- locate
- vi
- df
- du
- clear
- chmod
- tar
- .bashrc
- .bashprofile
- Scripts

Ficheros de configuración más importantes del directorio del usuario

.bash_profile

- “Script de login” privado del usuario (para el empleo del bash, variables de entorno o librerías del usuario). Se ejecuta cuando se inicia sesión

.bashrc

- Archivos de configuración del bash (se ejecuta en terminales)

.bash_history

- Archivos de protocolo del bash (historial del usuario)

Definir variables para nuestras rutas,
crear alias,
Ejecutar comandos

La diferencia entre `bashrc` y `bash_profile`

Los comandos `bash_profile` se ejecutan en non-login shell, es decir vía SSH, por consola, cada vez que nos logueamos y los de `bashrc` se ejecutan en cada instancia de bash, por ejemplo cada vez que abrimos una ventana de xterm.

¿para que usaremos el `bash_profile`?

cuando queramos que tal comando se ejecute solamente cuando inicie la maquina por primera vez o en una sección remota, un ejemplo es que un comando genere el estado del sistema cada vez que se inicie la maquina con nuestro usuario.

¿Cuándo usamos `bashrc`?

cada vez que le ejecutemos la terminal se correrá ese proceso muchas veces no es lo que queremos.

Comandos informativos

- logname
- hostname
- w
- whatis
- top
- ps
- ldd
- which
- whereis
- who
- whoami

Variables de ambiente

- Variables que contienen valores que pueden ser utilizadas por diferentes programas.
- env
- export
- echo
- PATH
- LIBRARY_PATH
- LD_LIBRARY_PATH

Módulos

module avail	Lista los módulos disponibles
module list	Lista los módulos cargados
module load nombre_módulo	Carga un modulo
module unload nombre_módulo	Remueve un módulo
module display nombre_módulo	Muestra las variables que carga el módulo
module purge	Quita los módulos cargados