

Interpretes de comandos

cat /etc/shells -> Los distintos interpretes de comandos que tengo disponible para ejecutar mi shell

ShortCuts

Flecha arriba/abajo -> Muestra el historial de comandos

Doble tab -> Muestra la cantidad de comandos que existen con las letras ingresadas

CTRL+R -> Te permite buscar un comando en el "historial" que guarda shell

cat .bashrc -> Muestra la configuración del usuario y del historial de comandos que utilizamos en bash. Existe en mi home.

```
# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
```

```
HISTSIZE=1000
```

```
HISTFILESIZE=2000
```

Mirar el resto de comandos en la diapositiva

Comandos Shell-Bash

nano "nombre.sh" -> Crea/abre el ejecutable de nuestro programa en bash. (En las comillas el nombre del archivo)

#!/bin/bash -> Le digo que tipo de shell voy a utilizar/interpretar el código. (En las comillas el PATH de la shell)

chmod +x "programa" -> Le agrego el permiso X (ejecucion) a todos los grupos de linux.

Por reglas de estilo las variables las definimos en MAYUSCULAS

Comandos varios y anotaciones

env o printenv -> Me permite ver las variables que estan definidas.

PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin -> Los ":" cambian la ruta de busqueda del PATH. El PATH permite ejecutar comandos como el autocompletar del TAB.

PATH=\$PATH -> ":/home/luciano/..." Agrego un directorio temporal para el \$PATH

Definicion de Path: It specifies the directories in which executable programs* are located on the machine that can be started without knowing and typing the whole path to the file on the command line. (<https://superuser.com/questions/284342/what-are-path-and-other-environment-variables-and-how-can-i-set-or-use-them>)

echo -n "Introduzca su nombre" -> el -n permite omitir el salto de linea cuando ingresa el dato un usuario.

```
if [ $USER == "root" ]; then
    echo "Sos root"
else
    echo "Solo lo podes ejecutar como root"
fi
```

Los if y elif los cierro con un fi

MUY IMPORTANTE LOS ESPACIOS ENTRE CONDICIONES

El == es SOLO para comparación entre caracteres

Los operadores de comparación entre números están en la diapositiva

En la guía hay distintos operadores para comparar el tamaño de archivos, si es un archivo de 0 byte, etc.

sh -x ./"Programa" depura de errores el programa, no es una forma muy "linda"

GNU nano 4.8

prog07.sh

Modificado

#!/bin/bash

Distintos tipos de programas y anotaciones

Me permite ejecutar un comando el operador `
for LINEA in `cat /etc/passwd`; do
echo \$LINEA
done

Si le doy la condicion IFS=\$'\n' estoy omitiendo el enter.

```
#!/bin/bash
IFS=$'\n'
for LINEA in `cat /etc/passwd`; do
echo $LINEA
done
```

```
#!/bin/bash
```

```
while read LINEA; do
echo $LINEA
done < /etc/passwd
Otra manera para que no lea el \n
```