

Informática II

La shell de Linux

Gonzalo F. Pérez Paina



Universidad Tecnológica Nacional
Facultad Regional Córdoba
UTN-FRC

– 2018 –

El shell de Linux

Introducción

¿Qué es una shell?

Es un programa que sirve de interfaz entre el usuario y el SO Linux. Permite introducir comandos y que el SO los ejecute.

El shell de Linux

Introducción

¿Qué es una shell?

Es un programa que sirve de interfaz entre el usuario y el SO Linux. Permite introducir comandos y que el SO los ejecute.

Originalmente Linux no disponía de interfaz gráfica

El shell de Linux

Introducción

¿Qué es una shell?

Es un programa que sirve de interfaz entre el usuario y el SO Linux. Permite introducir comandos y que el SO los ejecute.

Originalmente Linux no disponía de interfaz gráfica

Programas de shell

- ▶ Se puede programar rápidamente y de forma simple
- ▶ Disponible en la mayoría de las instalaciones del SO Linux
- ▶ Los programas de la shell se llaman *scripts* (interpretados en tiempo de ejecución)

El shell de Linux

Introducción

¿Qué es una shell?

Es un programa que sirve de interfaz entre el usuario y el SO Linux. Permite introducir comandos y que el SO los ejecute.

Originalmente Linux no disponía de interfaz gráfica

Programas de shell

- ▶ Se puede programar rápidamente y de forma simple
- ▶ Disponible en la mayoría de las instalaciones del SO Linux
- ▶ Los programas de la shell se llaman *scripts* (interpretados en tiempo de ejecución)

POSIX 1003.2 indica las especificaciones mínimas de una shell

El shell de Linux

Introducción

¿Qué es una shell?

Es un programa que sirve de interfaz entre el usuario y el SO Linux. Permite introducir comandos y que el SO los ejecute.

Originalmente Linux no disponía de interfaz gráfica

Programas de shell

- ▶ Se puede programar rápidamente y de forma simple
- ▶ Disponible en la mayoría de las instalaciones del SO Linux
- ▶ Los programas de la shell se llaman *scripts* (interpretados en tiempo de ejecución)

POSIX 1003.2 indica las especificaciones mínimas de una shell

Filosofía Linux

Utilidades simples y pequeñas que se utilizan como un eslabón de una cadena de comandos (`$ ls -la | more`)

El shell de Linux

Introducción

¿Qué es una shell?

Es un programa que sirve de interfaz entre el usuario y el SO Linux. Permite introducir comandos y que el SO los ejecute.

Originalmente Linux no disponía de interfaz gráfica

Programas de shell

- ▶ Se puede programar rápidamente y de forma simple
- ▶ Disponible en la mayoría de las instalaciones del SO Linux
- ▶ Los programas de la shell se llaman *scripts* (interpretados en tiempo de ejecución)

POSIX 1003.2 indica las especificaciones mínimas de una shell

Filosofía Linux

Utilidades simples y pequeñas que se utilizan como un eslabón de una cadena de comandos (\$ `ls -la | more`)

KISS: Keep It Small and Simple...o...

El shell de Linux

Introducción

¿Qué es una shell?

Es un programa que sirve de interfaz entre el usuario y el SO Linux. Permite introducir comandos y que el SO los ejecute.

Originalmente Linux no disponía de interfaz gráfica

Programas de shell

- ▶ Se puede programar rápidamente y de forma simple
- ▶ Disponible en la mayoría de las instalaciones del SO Linux
- ▶ Los programas de la shell se llaman *scripts* (interpretados en tiempo de ejecución)

POSIX 1003.2 indica las especificaciones mínimas de una shell

Filosofía Linux

Utilidades simples y pequeñas que se utilizan como un eslabón de una cadena de comandos (`$ ls -la | more`)

KISS: Keep It Small and Simple...o... (Keep It Simple Stupid :)

El shell de Linux

Usuarios y comandos

Linux es un SO tipo Unix, multiplataforma, multitarea y multiusuario.

El shell de Linux

Usuarios y comandos

Linux es un SO tipo Unix, multiplataforma, multitarea y multiusuario.

Usuario

- ▶ Para usar el SO es necesario abrir una sesión de trabajo (identificarse). Nro. de usuario, UID. `whoami`
- ▶ Los usuarios se organizan en grupos. Nro. de grupo, GID. `groups`. `id`
- ▶ Usuario **root** (superusuario o admin), cuenta con todos los privilegios

El shell de Linux

Usuarios y comandos

Linux es un SO tipo Unix, multiplataforma, multitarea y multiusuario.

Usuario

- ▶ Para usar el SO es necesario abrir una sesión de trabajo (identificarse). Nro. de usuario, UID. `whoami`
- ▶ Los usuarios se organizan en grupos. Nro. de grupo, GID. `groups. id`
- ▶ Usuario **root** (superusuario o admin), cuenta con todos los privilegios

Comandos

- ▶ Archivo binarios o ejecutables localizados en el sistema. `which`
Ej. `which whoami`. Variable `PATH`
- ▶ Auto-completar `TAB`

El shell de Linux

Usuarios y comandos

Linux es un SO tipo Unix, multiplataforma, multitarea y multiusuario.

Usuario

- ▶ Para usar el SO es necesario abrir una sesión de trabajo (identificarse). Nro. de usuario, UID. `whoami`
- ▶ Los usuarios se organizan en grupos. Nro. de grupo, GID. `groups. id`
- ▶ Usuario **root** (superusuario o admin), cuenta con todos los privilegios

Comandos

- ▶ Archivo binarios o ejecutables localizados en el sistema. `which`
Ej. `which whoami`. Variable `PATH`
- ▶ Auto-completar `TAB`

- ▶ `/bin`: binarios utilizados por el sistema de arranque/buteo
- ▶ `/usr/bin`: binarios de usuario
- ▶ `/usr/local/bin`: binarios locales, programas específicos de la instalación

El shell de Linux

Variables de entorno y... más comandos

Variables de entorno

- ▶ Variables inicializadas desde el entorno
- ▶ Escritas en mayúsculas para distinguirlas de otras variables (usuario)
- ▶ Ej.: \$HOME, \$PATH, \$PS1, \$PS2, \$IFS, \$0, \$#, \$\$
- ▶ Imprimir valor de variable de entorno: `echo $HOME`

El shell de Linux

Variables de entorno y... más comandos

Variables de entorno

- ▶ Variables inicializadas desde el entorno
- ▶ Escritas en mayúsculas para distinguirlas de otras variables (usuario)
- ▶ Ej.: \$HOME, \$PATH, \$PS1, \$PS2, \$IFS, \$0, \$#, \$\$
- ▶ Imprimir valor de variable de entorno: `echo $HOME`

Más sobre comandos:

- ▶ Opciones de los comandos, letra seguido de '-'. O bien '--' (`--help`, `--version`)

El shell de Linux

Variables de entorno y... más comandos

Variables de entorno

- ▶ Variables inicializadas desde el entorno
- ▶ Escritas en mayúsculas para distinguirlas de otras variables (usuario)
- ▶ Ej.: \$HOME, \$PATH, \$PS1, \$PS2, \$IFS, \$0, \$#, \$\$
- ▶ Imprimir valor de variable de entorno: `echo $HOME`

Más sobre comandos:

- ▶ Opciones de los comandos, letra seguido de '-'. O bien '--' (`--help`, `--version`)
- ▶ Comandos `echo`, `printenv`, `man`, `apropos`

El shell de Linux

Variables de entorno y... más comandos

Variables de entorno

- ▶ Variables inicializadas desde el entorno
- ▶ Escritas en mayúsculas para distinguirlas de otras variables (usuario)
- ▶ Ej.: \$HOME, \$PATH, \$PS1, \$PS2, \$IFS, \$0, \$#, \$\$
- ▶ Imprimir valor de variable de entorno: `echo $HOME`

Más sobre comandos:

- ▶ Opciones de los comandos, letra seguido de '-'. O bien '--' (`--help`, `--version`)
- ▶ Comandos `echo`, `printenv`, `man`, `apropos`
- ▶ **shell prompt**: aparece en la línea de comandos indicando que está a la espera de órdenes (`$ echo $SHELL`, `$ bash --version`)

El shell de Linux

Variables de entorno y... más comandos

Variables de entorno

- ▶ Variables inicializadas desde el entorno
- ▶ Escritas en mayúsculas para distinguirlas de otras variables (usuario)
- ▶ Ej.: \$HOME, \$PATH, \$PS1, \$PS2, \$IFS, \$0, \$#, \$\$
- ▶ Imprimir valor de variable de entorno: `echo $HOME`

Más sobre comandos:

- ▶ Opciones de los comandos, letra seguido de '-'. O bien '--' (`--help`, `--version`)
- ▶ Comandos `echo`, `printenv`, `man`, `apropos`
- ▶ **shell prompt**: aparece en la línea de comandos indicando que está a la espera de órdenes (`$ echo $SHELL`, `$ bash --version`)
- ▶ **Manpages**: manual en línea (RTFM).
`$ man printenv`
`$ man 1 printf`
`$ man 3 printf`
`$ man man`

El shell de Linux

Variables de entorno y... más comandos

Variables de entorno

- ▶ Variables inicializadas desde el entorno
- ▶ Escritas en mayúsculas para distinguirlas de otras variables (usuario)
- ▶ Ej.: \$HOME, \$PATH, \$PS1, \$PS2, \$IFS, \$0, \$#, \$\$
- ▶ Imprimir valor de variable de entorno: `echo $HOME`

Más sobre comandos:

- ▶ Opciones de los comandos, letra seguido de '-'. O bien '--' (`--help`, `--version`)
- ▶ Comandos `echo`, `printenv`, `man`, `apropos`
- ▶ **shell prompt**: aparece en la línea de comandos indicando que está a la espera de órdenes (`$ echo $SHELL`, `$ bash --version`)
- ▶ **Manpages**: manual en línea (RTFM).
`$ man printenv`
`$ man 1 printf`
`$ man 3 printf`
`$ man man`

Variable \$PATH y comando `export` (`$ man export`)

El shell de Linux

Páginas de manuales (manpages)

Cuenta con diferentes secciones: `$ man man`

El shell de Linux

Páginas de manuales (manpages)

Cuenta con diferentes secciones: `$ man man`

Algunas son:

1. Executable programs or shell commands
2. System calls (functions provided by the kernel)
3. Library calls (functions within program libraries)
4. Special files (usually found in `/dev`)
5. File formats and conventions eg. `/etc/passwd`
6. Games
7. Miscellaneous (including macro packages and conventions)
8. System administration commands (usually only for root)

El shell de Linux

Páginas de manuales (manpages)

Cuenta con diferentes secciones: `$ man man`

Algunas son:

1. Executable programs or shell commands
2. System calls (functions provided by the kernel)
3. Library calls (functions within program libraries)
4. Special files (usually found in `/dev`)
5. File formats and conventions eg. `/etc/passwd`
6. Games
7. Miscellaneous (including macro packages and conventions)
8. System administration commands (usually only for root)

Ejemplos: `$ man 1 printf`, `$ man 3 printf`, `$ man -a printf`
`$ man -k '^printf'`

El shell de Linux

Sistema de archivos y permisos

- ▶ Estructura de archivos en árbol. Archivos “tipo directorio”. Comando `ls`

El shell de Linux

Sistema de archivos y permisos

- ▶ Estructura de archivos en árbol. Archivos “tipo directorio”. Comando `ls`
- ▶ Directorios (raíz/root), camino/path. Directorio actual (`.`), anterior o padre (`..`)

El shell de Linux

Sistema de archivos y permisos

- ▶ Estructura de archivos en árbol. Archivos “tipo directorio”. Comando `ls`
- ▶ Directorios (raíz/root), camino/path. Directorio actual (`.`), anterior o padre (`..`)
Ver archivos ocultos: `$ ls -a`

El shell de Linux

Sistema de archivos y permisos

- ▶ Estructura de archivos en árbol. Archivos “tipo directorio”. Comando `ls`
- ▶ Directorios (raíz/root), camino/path. Directorio actual (`.`), anterior o padre (`..`)
Ver archivos ocultos: `$ ls -a`
- ▶ Directorio `/home`. Variable de entorno `HOME` (`ls $HOME`). Comandos `cd`, `pwd`

El shell de Linux

Sistema de archivos y permisos

- ▶ Estructura de archivos en árbol. Archivos “tipo directorio”. Comando `ls`
- ▶ Directorios (raíz/root), camino/path. Directorio actual (`.`), anterior o padre (`..`)
Ver archivos ocultos: `$ ls -a`
- ▶ Directorio `/home`. Variable de entorno `HOME` (`ls $HOME`). Comandos `cd`, `pwd`
- ▶ Camino absoluto (comienza en `/`), y camino relativo (comienza en `./` o `../`)

El shell de Linux

Sistema de archivos y permisos

- ▶ Estructura de archivos en árbol. Archivos “tipo directorio”. Comando `ls`
- ▶ Directorios (raíz/root), camino/path. Directorio actual (`.`), anterior o padre (`..`)
Ver archivos ocultos: `$ ls -a`
- ▶ Directorio `/home`. Variable de entorno `HOME` (`ls $HOME`). Comandos `cd`, `pwd`
- ▶ Camino absoluto (comienza en `/`), y camino relativo (comienza en `./` o `../`)
- ▶ Comandos `touch`, `rm`, `mkdir`, `rmdir`, `cp`, `mv`. (`$ which cd`)

El shell de Linux

Sistema de archivos y permisos

- ▶ Estructura de archivos en árbol. Archivos “tipo directorio”. Comando `ls`
- ▶ Directorios (raíz/root), camino/path. Directorio actual (`.`), anterior o padre (`..`)
Ver archivos ocultos: `$ ls -a`
- ▶ Directorio `/home`. Variable de entorno `HOME` (`ls $HOME`). Comandos `cd`, `pwd`
- ▶ Camino absoluto (comienza en `/`), y camino relativo (comienza en `./` o `../`)
- ▶ Comandos `touch`, `rm`, `mkdir`, `rmdir`, `cp`, `mv`. (`$ which cd`)
- ▶ Atributos de archivos (`$ ls -l /`)

El shell de Linux

Sistema de archivos y permisos

- ▶ Estructura de archivos en árbol. Archivos “tipo directorio”. Comando `ls`
- ▶ Directorios (raíz/root), camino/path. Directorio actual (`.`), anterior o padre (`..`)
Ver archivos ocultos: `$ ls -a`
- ▶ Directorio `/home`. Variable de entorno `HOME` (`ls $HOME`). Comandos `cd`, `pwd`
- ▶ Camino absoluto (comienza en `/`), y camino relativo (comienza en `./` o `../`)
- ▶ Comandos `touch`, `rm`, `mkdir`, `rmdir`, `cp`, `mv`. (`$ which cd`)
- ▶ Atributos de archivos (`$ ls -l /`)

Hacer:

```
$ cd
$ touch hola.txt
$ ls -l hola.txt
```


El shell de Linux

Sistema de archivos y permisos

Alterar permisos de archivos. Comando `chmod`

- ▶ `$ chmod u-r hola.txt.`

El shell de Linux

Sistema de archivos y permisos

Alterar permisos de archivos. Comando `chmod`

- ▶ `$ chmod u-r hola.txt.`

- ▶ `$ cat hola.txt`

El shell de Linux

Sistema de archivos y permisos

Alterar permisos de archivos. Comando `chmod`

- ▶ `$ chmod u-r hola.txt.`
- ▶ `$ cat hola.txt`
- ▶ `$ chmod -x a.out`

El shell de Linux

Sistema de archivos y permisos

Alterar permisos de archivos. Comando `chmod`

- ▶ `$ chmod u-r hola.txt.`
- ▶ `$ cat hola.txt`
- ▶ `$ chmod -x a.out`
- ▶ `$ chmod o+x hola_bash.sh` (Crear script Shell)

El shell de Linux

Sistema de archivos y permisos

Alterar permisos de archivos. Comando `chmod`

- ▶ `$ chmod u-r hola.txt.`
- ▶ `$ cat hola.txt`
- ▶ `$ chmod -x a.out`
- ▶ `$ chmod o+x hola_bash.sh` (Crear script Shell)

Notación numérica de los permisos

`r w x | - w x | r - x`
`4 2 1 | 0 2 1 | 4 0 1`

Equivale a un permiso 735
 $4+2+1, 0+2+1, 4+0+1 = 7,3,5$

El shell de Linux

Más comandos :)

- ▶ `uname (-n, -v, -r, -m, -a)`
- ▶ `lshw, lsusb, lspci (-tv)`
- ▶ `uptime`
- ▶ `df -h`
- ▶ `dmesg`
- ▶ `grep`

El shell de Linux

Redirección de entrada y salida. Pipes

Redirección de salida

- ▶ `$ ls -l > lsoutput.txt`

Redirecciona la salida del comando `ls` al archivo `lsoutput.txt`

El shell de Linux

Redirección de entrada y salida. Pipes

Redirección de salida

- ▶ `$ ls -l > lsoutput.txt`
Redirecciona la salida del comando `ls` al archivo `lsoutput.txt`
- ▶ `$ ps >> lsoutput.txt`
Agrega la salida del comando `ps` al archivo

El shell de Linux

Redirección de entrada y salida. Pipes

Redirección de salida

- ▶ `$ ls -l > lsoutput.txt`
Redirecciona la salida del comando `ls` al archivo `lsoutput.txt`
- ▶ `$ ps >> lsoutput.txt`
Agrega la salida del comando `ps` al archivo
- ▶ `$ kill -9 1234 > killout.txt 2> killerr.txt`

El shell de Linux

Redirección de entrada y salida. Pipes

Redirección de salida

- ▶ `$ ls -l > lsoutput.txt`
Redirecciona la salida del comando `ls` al archivo `lsoutput.txt`
- ▶ `$ ps >> lsoutput.txt`
Agrega la salida del comando `ps` al archivo
- ▶ `$ kill -9 1234 > killout.txt 2> killerr.txt`
- ▶ `$ kill -1 1234 > killouterr.txt 2>&1`

El shell de Linux

Redirección de entrada y salida. Pipes

Redirección de salida

- ▶ `$ ls -l > lsoutput.txt`
Redirecciona la salida del comando `ls` al archivo `lsoutput.txt`
- ▶ `$ ps >> lsoutput.txt`
Agrega la salida del comando `ps` al archivo
- ▶ `$ kill -9 1234 > killout.txt 2> killerr.txt`
- ▶ `$ kill -1 1234 > killouterr.txt 2>&1`
- ▶ `$ kill -1 1234 > /dev/null 2>&1`

(Descriptor de archivo 0: entrada estándar, 1: salida estándar, 2: salida de error estándar)

El shell de Linux

Redirección de entrada y salida. Pipes

Redirección de salida

- ▶ `$ ls -l > lsoutput.txt`
Redirecciona la salida del comando `ls` al archivo `lsoutput.txt`
- ▶ `$ ps >> lsoutput.txt`
Agrega la salida del comando `ps` al archivo
- ▶ `$ kill -9 1234 > killout.txt 2> killerr.txt`
- ▶ `$ kill -1 1234 > killouterr.txt 2>&1`
- ▶ `$ kill -1 1234 > /dev/null 2>&1`

(Descriptor de archivo 0: entrada estándar, 1: salida estándar, 2: salida de error estándar)

Redirección de entrada

`[$ ls -lR /usr/include > lsoutput.txt]`

- ▶ `$ more < lsoutput.txt`

El shell de Linux

Redirección de entrada y salida. Pipes

Se puede conectar procesos utilizando el operador pipe, |. Ejemplo: utilizar `sort` para ordenar la salida de `ps`

El shell de Linux

Redirección de entrada y salida. Pipes

Se puede conectar procesos utilizando el operador pipe, |. Ejemplo: utilizar `sort` para ordenar la salida de `ps`

Si no se utiliza pipes, se necesitan varios pasos

1. `$ ps > psout.txt`
2. `$ sort psout.txt > pssort.txt`

El shell de Linux

Redirección de entrada y salida. Pipes

Se puede conectar procesos utilizando el operador pipe, |. Ejemplo: utilizar `sort` para ordenar la salida de `ps`

Si no se utiliza pipes, se necesitan varios pasos

1. `$ ps > psout.txt`
2. `$ sort psout.txt > pssort.txt`

Conectando los procesos mediante pipes

- ▶ `$ ps | sort > pssort.txt`

El shell de Linux

Redirección de entrada y salida. Pipes

Se puede conectar procesos utilizando el operador pipe, |. Ejemplo: utilizar `sort` para ordenar la salida de `ps`

Si no se utiliza pipes, se necesitan varios pasos

1. `$ ps > psout.txt`
2. `$ sort psout.txt > pssort.txt`

Conectando los procesos mediante pipes

- ▶ `$ ps | sort > pssort.txt`

Otros ejemplos

- ▶ `$ ps aux | sort`
- ▶ `$ ps aux | sort | more`

El shell de Linux

Procesos

- ▶ En un sistema monotarea se utiliza generalmente el término programa.
- ▶ En un sistema multitarea se utiliza el término proceso, indicando que el mismo está arrancado y en funcionamiento.
- ▶ Un programa puede dar lugar a varios procesos.
- ▶ Un proceso puede estar detenido (dormido), pero existe información del estado del mismo.
- ▶ Cada proceso tiene un nro. que lo identifica, PID, PPID
- ▶ Comandos `ps`, `top`/`htop`, `kill` (señal 15 y 9)
- ▶ `$ ps -p`, `$ ps axf`

El shell de Linux

Procesos

- ▶ En un sistema monotarea se utiliza generalmente el término programa.
- ▶ En un sistema multitarea se utiliza el término proceso, indicando que el mismo está arrancado y en funcionamiento.
- ▶ Un programa puede dar lugar a varios procesos.
- ▶ Un proceso puede estar detenido (dormido), pero existe información del estado del mismo.
- ▶ Cada proceso tiene un nro. que lo identifica, PID, PPID
- ▶ Comandos `ps`, `top`/`htop`, `kill` (señal 15 y 9)
- ▶ `$ ps -p`, `$ ps axf`

Atributos de los procesos

- ▶ **PID**: Valor numérico que identifica al proceso
- ▶ **TTY**: Terminal asociada al proceso
- ▶ **STAT**: Estado del proceso
- ▶ **TIME**: Tiempo de CPU consumido por el proceso
- ▶ **COMMAND**: Comandos y argumentos utilizados

Bibliografía

- ▶ *Linux Programming Unleashed*, 2nd Edition, SAMS (1999)
- ▶ *Beginning Linux Programming*, 4th Edition, Wiley (2007)
- ▶ *Curso de Linux para novatos, brutos y extremadamente torpes*, Ed. nov. 2008
- ▶ Manuales y tutoriales de la web