Introducción a GNU/Linux

20/08/2015

¿Qué es un Sistema Operativo (SO)?

SO: es el software que se ejecuta al iniciar la computadora y que posibilita el manejo de la misma.

funciones básicas

- Provee interfaces al usuario.
- Gestiona los procesos.
- Gestiona la memoria principal y el almacenamiento secundario.
- Administrar el uso de los dispositivos de E/S.
- Administra los archivos.
- Provee un sistema de protección.

¿Qué es GNU/Linux?

Es un sistema operativo (y un conjunto de aplicaciones) con las siguientes características:

1. Multitarea
2. Multiusuario
3. Estable
4. Seguro
5. Libre
6. Con soporte (comunidad + empresas)

- El software nació libre.
- En los 70 las companías comiezan a imponer restricciones a los usuarios (licencias).
- En los 80 ..

  Richard M. Stallman, y la famosa impresora ...

  Dilema: $\neg$ comparto $\vee$ violo la licencia

"Podría haber hecho dinero de esta manera, y tal vez me hubiese divertido escribiendo código. Pero sabía que al final de mi carrera, al mirar los años que pasé construyendo paredes para dividir a la gente, sentiría que usé mi vida para empeorar el mundo."

En lugar de convencer a las empresas pensó en producir los mismos programas pero **libres**. Lo que primero necesitaba era un SO libre.

- En 1983 R. Stallman funda el movimiento GNU.
- En 1984 funda "Free Software Fundation".
- Hacia 1991, GNU había creado:
  - Licencia GPL.
  - Aplicaciones libres

  En el mismo año **Linus Torvalds** tomando como partida el SO Minix escribe un kernel para la plataforma intel x386.
- La unión del SO GNU al que sólo le faltaba un kernel, con el kernel diseñado por Linus dió lugar al SO **GNU/Linux**.

Licencias

- El documento que especifica qué libertades se le otorgan y cuáles se le niegan a los usuarios se denomina **licencia**.
- Una **licencia es libre** si otorga la libertad de:

Libertad 0 ejecutar el programa para cualquier propósito.
Libertad 1 estudiar cómo funciona y modificarlo
Libertad 2 redistribuir copias y así ayudar a tu prójimo.
Libertad 3 contribuir a la comunidad: hacer y distribuir mejoras.

- **GPL** es la licencia de software libre más utilizada.
- Las compañías que venden software no libre (bajo la licencia propietaria) no venden software sino "el permiso" para usar software.

¿Por qué enseñamos SL en la Universidad?

- Costos, no es necesario pagar licencias.
- No condicionamos al estudiante a pagar licencias para usar en su casa el software que le enseñamos.
- El software puede adaptarse y modificarse localmente.
- La solidaridad y el compartir son actitudes socialmente positivas también en el mundo del software.
- Es una forma de reconstruir la infraestructura social que se ha perdido en el tiempo con sistemas individualistas.

Distribuciones (o distros) de GNU/Linux

Variantes de GNU/Linux diseñadas para satisfacer las necesidades de un grupo específico de usuarios. Algunas de ellas son para:

- usuarios principiantes (Ubuntu)
- usuarios curiosos (Arch)
- edición de audio, imágenes y video (JAD, Musix)
- para equipos antiguos o con pocos recursos (Puppy Linux, pesa 130 MiB).
- educación (Qimo, EduLinux, edubuntu)
- etc.

Terminología GNU/Linux

- **Shell, consola o terminal** (intérprete de comandos): Interpreta los comandos introducidos por el usuario y los pasa al SO.
- **Comand Line** (línea de comandos): Lugar donde se escriben los comandos en el shell. Ej: [shrek@pantano:]$
- **Root o superusuario**: La cuenta root se crea durante la instalación y tiene acceso completo al sistema. Consejo: Usar lo menos posible el usuario root!
- **Graphical User Interface (GUI)**: Interfaz de software gráfica. Un *desktop environment* (entorno de escritorio) provee los elementos para crear una GUI. Los como ser GNOME, KDE, Unity, i3, etc.

Comandos shell

- La mayoría de los comandos requieren de opciones.
- Las páginas de manual dan información detallada sobre un comando y sus opciones.
- Se puede hacer que el shell complete la línea de comandos con la tecla Tab.

Comandos más usados

man	Comando para aprender a usar los comandos
pwd	Muestra el directorio actual
ls	Lista el contenido de un directorio
cd	Permite moverse a través del sistema de directorios
cat	Imprime en patalla el contenido de un archivo
grep	Permite filtrar
mkdir	Crea un directorio
rmdir	Borra archivos y directorios vacíos (la opción -r permite borrar directorios llenos)
cp	Realiza copia de archivos.
rm	Borra un archivo de un directorio
file	Muestra el tipo de archivo y su nombre
find	Busca archivos y directorios.

Más comandos

ps	Muestra los procesos que se están ejecutando en el momento.
kill	Envía señales a los procesos.
du	Muestra el espacio que ocupan los archivos en el disco
passwd	Para cambiar la contraseña del usuario actual.
exit/logout	Cierra la sesión.

Caracteres “comodines”

Se pueden utilizar en los intérpretes de comandos.

- El comodín * hace referencia a cualquier cadena de caracteres.

  Ejemplos:

  ```
  ls a*
  cp /etc/*.conf /home/mihome/configuraciones
  mv *.hs practicas/
  ```

- El comodín ? hace referencia a un sólo caracter.

  Ejemplos:

  ```
  ls ?ipo
  cat ??libro*.txt
  ```

Permisos

Al ser Linux/Unix sistemas multiusuarios, es necesario que el sistema provea un mecanismo que permita decidir de quién es cada cosa.

La seguridad en Linux se centra en que cada archivo del sistema tiene:

- Un dueño
- Un grupo
- Permisos de archivo

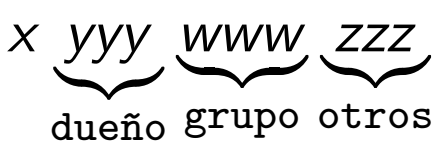
Los permisos están divididos en 3 tipos:

- lectura (se indica con la letra r),
- escritura (w) y
- ejecución (x).

Permisos

Cuando ejecutamos el comando ls -l, la primer columna corresponde al permiso del archivo o directorio.

Éstos están formados por 10 dígitos:



El primer dígito es una d si es un directorio o - si es un archivo.

Ejemplos: -rw-r- -r- - , -rw-rw-r- - , drw-rw-rw-, -rwxrw-rw-, -rwxrwxrwx , drwxrwxrwx

Hay dependencias!

Cambiar permisos

Con el comando chmod. Sintaxis:

chmod {a,u,g,o}{+,-}{r,w,x} nombre

- El primer argumento se refiere a quien le estamos dando o quitando el permiso. Puede ser 'a' (all),'u' (usuario), 'g' (grupo) y/o 'o' (otros).
- El segundo indica si se están añadiendo o quitando permisos.
- El tercero indica el tipo del permiso.

Ejemplos:

chmod go+rw archivo
chmod o-rwx archivo
chmod +x archivo

Entrada/Salida

Descriptores: STDIN(0), STDOUT(1), STDERR(2)

man	Comando para aprender a usar los comandos
pwd	Muestra el directorio actual
ls	Lista el contenido de un directorio
cd	Permite moverse a través del sistema de directorios
cat	Imprime en patalla el contenido de un archivo
grep	Permite filtrar
mkdir	Crea un directorio
rmdir	Borra archivos y directorios vacíos (la opción -r permite borrar directorios llenos)
cp	Realiza copia de archivos.
rm	Borra un archivo de un directorio
file	Muestra el tipo de archivo y su nombre
find	Busca archivos y directorios.

Más comandos

ps	Muestra los procesos que se están ejecutando en el momento.
kill	Envía señales a los procesos.
du	Muestra el espacio que ocupan los archivos en el disco
passwd	Para cambiar la contraseña del usuario actual.
exit/logout	Cierra la sesión.

Caracteres “comodines”

Se pueden utilizar en los intérpretes de comandos.

- El comodín * hace referencia a cualquier cadena de caracteres.

  Ejemplos:

  ```
  ls a*
  cp /etc/*.conf /home/mihome/configuraciones
  mv *.hs practicas/
  ```

- El comodín ? hace referencia a un sólo caracter.

  Ejemplos:

  ```
  ls ?ipo
  cat ??libro*.txt
  ```

Permisos

Al ser Linux/Unix sistemas multiusuarios, es necesario que el sistema provea un mecanismo que permita decidir de quién es cada cosa.

La seguridad en Linux se centra en que cada archivo del sistema tiene:

- Un dueño
- Un grupo
- Permisos de archivo

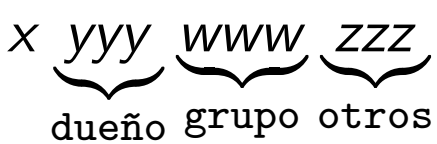
Los permisos están divididos en 3 tipos:

- lectura (se indica con la letra r),
- escritura (w) y
- ejecución (x).

Permisos

Cuando ejecutamos el comando ls -l, la primer columna corresponde al permiso del archivo o directorio.

Éstos están formados por 10 dígitos:



El primer dígito es una d si es un directorio o - si es un archivo.

Ejemplos: -rw-r- -r- - , -rw-rw-r- - , drw-rw-rw-, -rwxrw-rw-, -rwxrwxrwx , drwxrwxrwx

Hay dependencias!

Cambiar permisos

Con el comando chmod. Sintaxis:

chmod {a,u,g,o}{+,-}{r,w,x} nombre

- El primer argumento se refiere a quien le estamos dando o quitando el permiso. Puede ser 'a' (all),'u' (usuario), 'g' (grupo) y/o 'o' (otros).
- El segundo indica si se están añadiendo o quitando permisos.
- El tercero indica el tipo del permiso.

Ejemplos:

chmod go+rw archivo
chmod o-rwx archivo
chmod +x archivo

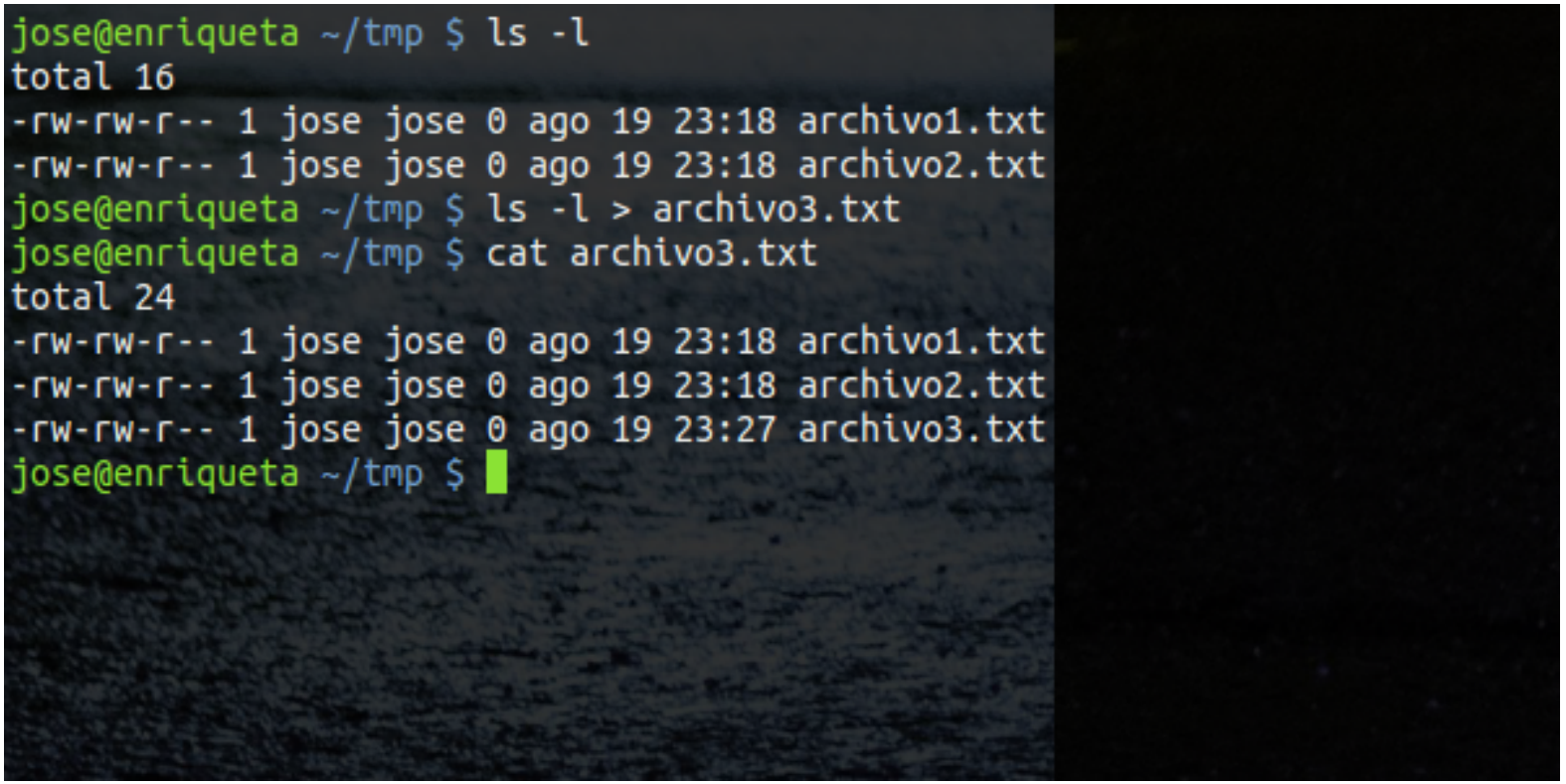
Entrada/Salida

Siempre tenemos tres archivos abiertos en una consola.

STDIN	Entrada estándar, teclado.
STDOUT	Salida estándar, la pantalla.
STDERR	Mensajes de error, también a la pantalla.

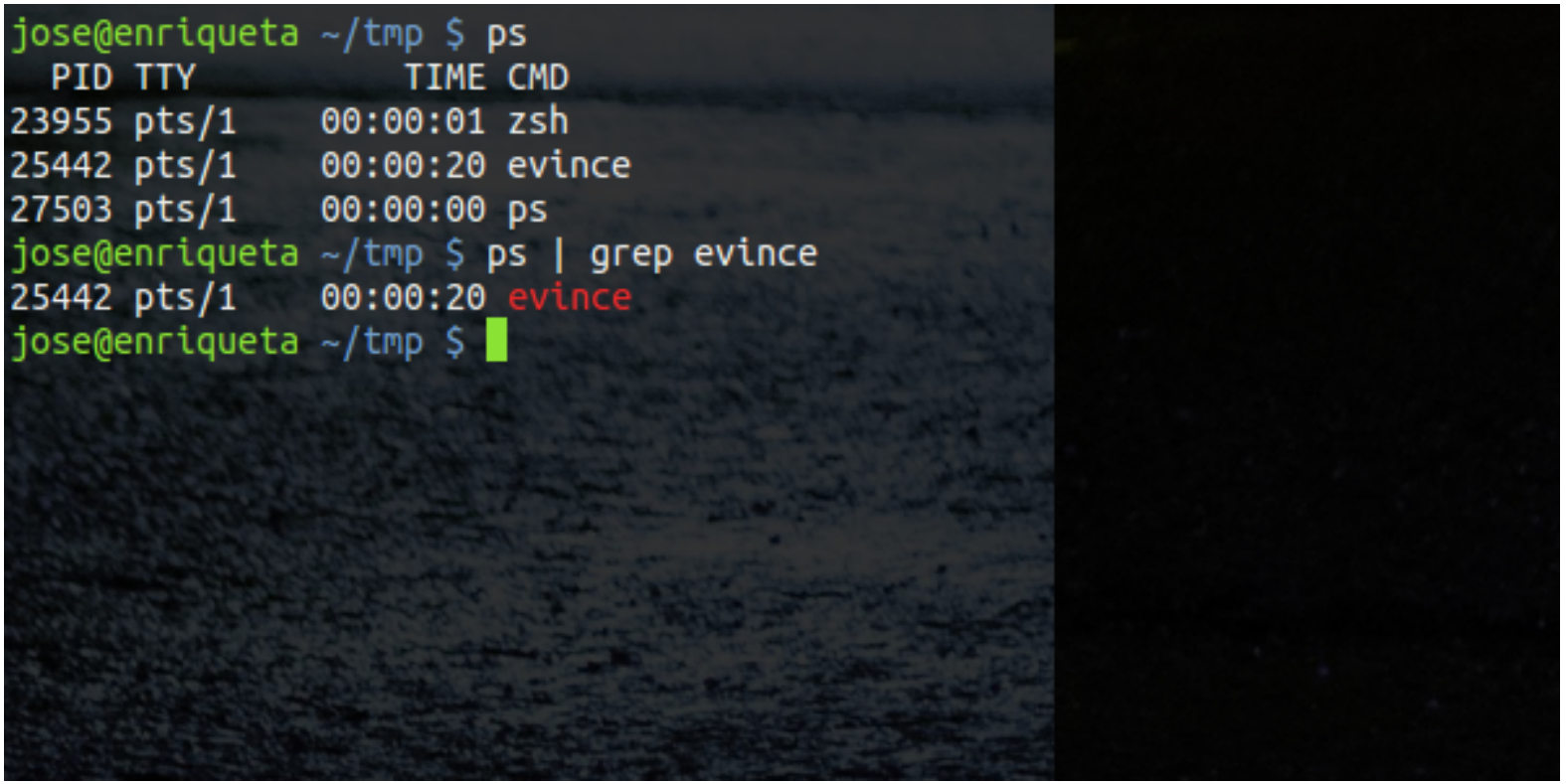
Los descriptores de archivos asociados a stdin, stdout y stderr son el 0, 1 y 2 respectivamente.

comando 1> *archivo* escribe la salida estándar de comando en archivo



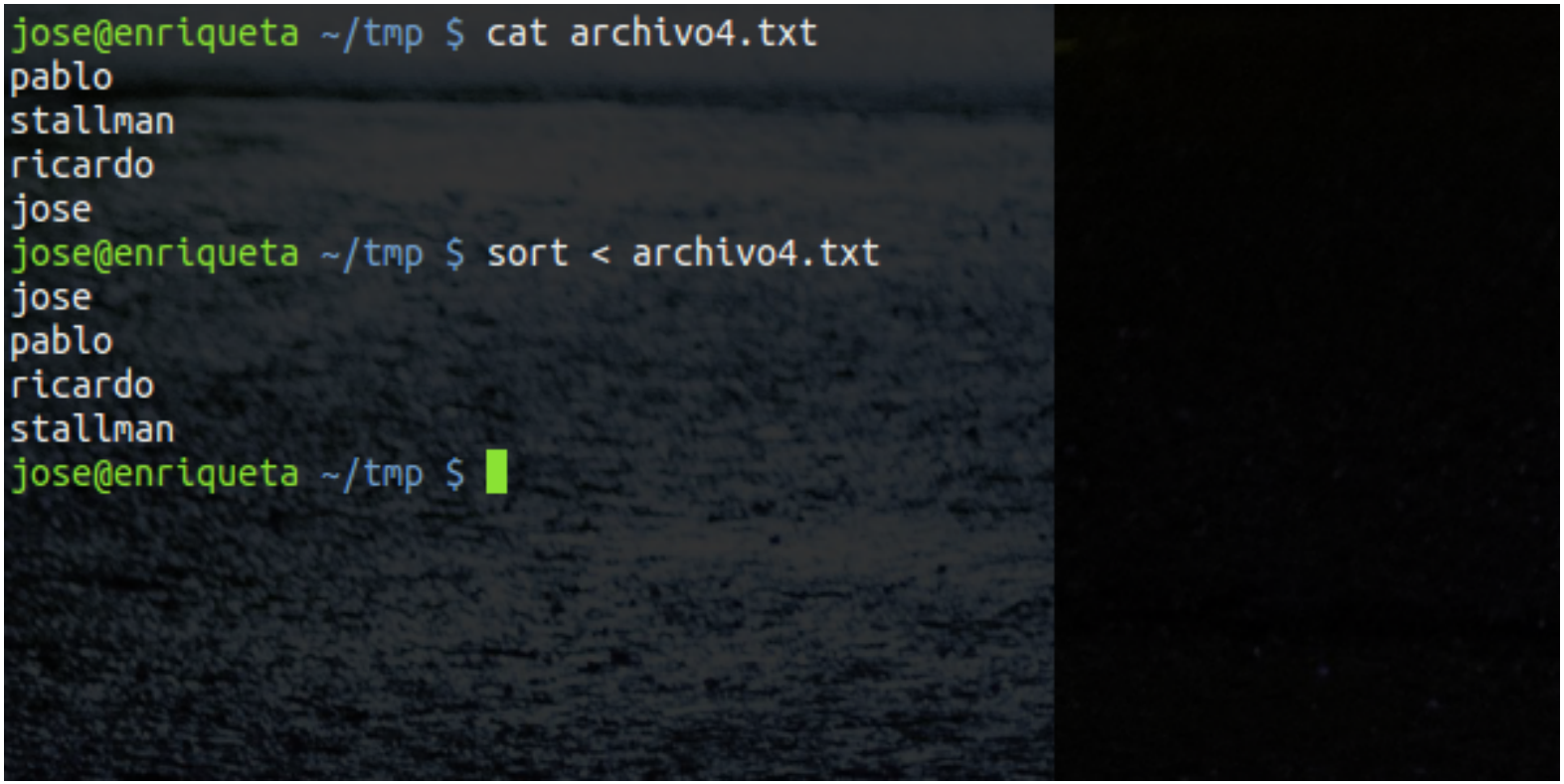
Entrada/Salida - Redirección

comando1 | *comando2* conecta la salida estándar del comando1 en la entrada estándar comando2



Entrada/Salida - Redirección

comando < *archivo* envía archivo a la entrada estándar de comando



Referencias


http://www.gnu.org/gnu/linux-and-gnu.es.html


M. Notti, M. Carr y G. Accardo. Charla dada a estudiantes de Icc sobre el *Software Libre*.