

PROGRAMACIÓN (E.U.I.). Curso 2001/2002
Práctica 1. LINUX, JAVA y HTML

F. Marqués, N. Prieto y J. Piris

Índice General

1	El sistema operativo LINUX	1
2	El editor de textos <i>EMACS</i>	2
3	Editar, compilar y ejecutar un programa en Java	3
4	JAVA y HTML	3

1 El sistema operativo LINUX

Un sistema operativo es un conjunto de programas que se encarga, entre otras cosas, de la gestión y ejecución de los programas de usuario, el control de las entradas y salidas, el mantenimiento de los ficheros y del control de errores tanto de programación como de los diferentes dispositivos.

Los comandos más utilizados son los siguientes:

1. Para lanzar el entorno de ventanas **startx**.
2. Para cambiar la contraseña **passwd**.
3. Comandos sobre directorios:
 - **pwd** muestra el camino absoluto desde la raíz al directorio actual de trabajo.
 - **ls** lista los ficheros y directorios de un directorio. Si no recibe ningún argumento el directorio por defecto es el de trabajo. Este comando con la opción **-l** muestra mas información sobre cada item y con la opción **-a** muestra los ficheros ocultos.
 - **mkdir NombreDirectorio** crea un directorio vacío de nombre *NombreDirectorio*
 - **cd NombreDirectorio** pasa del directorio actual al directorio hijo *NombreDirectorio*
 - **cd ..** cambia al directorio padre

- `rmdir NombreDirectorio` borra el directorio *NombreDirectorio* si está vacío
4. Comandos sobre ficheros:
 - `cp NombreFichOrigen NombreFichDestino` copia el contenido del fichero origen a destino
 - `mv origen destino` cambia el nombre del fichero *origen* que pasará a llamarse *destino*
 - `mv NombreFichero NombreDirectorio` cambia el fichero al directorio *NombreDirectorio*
 - `rm NombreFichero` borra el fichero indicado
 - `cat NombreFichero` visualiza el contenido del fichero
 - `more NombreFichero` visualiza el contenido del fichero página a página
 5. Comandos sobre redirecciones de la entrada y salida estándar:
 - `Comando > NombreFichDestino` redirecciona la salida estándar del comando al fichero *NombreFichDestino*
 - `Comando >> NombreFichDestino` igual que el anterior pero *añadiendo* la información al final del fichero
 - `Comando < NombreFichOrigen` redirecciona la entrada estándar del comando al fichero *NombreFichOrigen*
 - `Comando 2> NombreFichOrigen` redirecciona la salida de error del comando al fichero *NombreFichOrigen*
 - `Comando1 | Comando2` redirecciona la salida estándar del primer comando a la entrada estándar del segundo comando
 - `tee NombreFichDestino1 ... NombreFichDestinon` copia su entrada estándar en su salida estándar y en todos los ficheros pasados como argumentos
 6. Para imprimir un fichero: `imprimir NombreFich1 ... NombreFichn`
 7. Ayuda en línea: `man NombreComando` explica cómo funciona el comando *NombreComando*

2 El editor de textos *EMACS*

Un editor de textos sirve para crear, leer, escribir y modificar ficheros. Los comandos más utilizados son:

1. Llamar al editor: `emacs NombreFichero`
2. Salvar o guardar un documento: `Ctrl-x Ctrl-s`
3. Salir del editor: `Ctrl-x Ctrl-c`
4. Abortar la ejecución de un comando: `Ctrl-g`

3 Editar, compilar y ejecutar un programa en Java

Lo primero que haremos es llamar al editor:

```
emacs Hola.java &
```

En la ventana de edición copiaremos el siguiente programa:

```
class Hola
{
    public static void main (String args[])
    {
        System.out.println('‘Hola, hola’');
    }
}
```

A continuación guardamos el contenido del fichero y compilamos:

```
javac Hola.java
```

Si el programa no tiene errores de compilación, se genera un fichero con el mismo nombre que el que contiene el texto del programa pero con la extensión `.class`. Finalmente ya podemos ejecutar el programa:

```
java Hola
```

4 JAVA y HTML

Un *applet* es una aplicación Java diseñada para ejecutarse en el contexto de un navegador. De la misma manera que un navegador carga textos, gráficos y sonidos a partir de las indicaciones contenidas en un fichero *HTML* (Hyper Text Markup Language), puede cargar el código de un applet Java y lanzar su ejecución en respuesta a una marca específica contenida en el fichero HTML.

Vamos a empezar compilando y ejecutando un applet sencillo. Este es el código del programa *Hola* convertido en un applet. Escríbelo en un fichero que se llame *HolaApplet.java*.

```
import java.awt.*;
import java.applet.*;

public class HolaApplet extends Applet {
    public void paint(Graphics g) {
        g.drawString("Hola, hola",20,20);
    }
}
```

El programa comienza con dos líneas que importan todos los paquetes de `java.applet` y `java.awt`. El paquete AWT (Abstract Window Toolkit) contiene las herramientas para trabajar en modo gráfico. Aquí veremos sólo algunas cosas muy sencillas y dejaremos otras para más adelante. A continuación en el programa se sobrescribe el método `paint` de la clase `Applet` y se le llama con un objeto `Graphics`, que se utiliza para dibujar. A continuación escribe la cadena “Hola, hola” en la posición de la pantalla (20,20).

Para ejecutar esta aplicación en un visualizador de red, hay que escribir un fichero HTML. La etiqueta `<applet>` se utiliza para iniciar una applet. Se le indica que cargue la applet compilada de Java `HolaApplet.class`, que está en el mismo directorio que este documento HTML. El tamaño de la applet se establece a 300 pixels de anchura por 60 de altura.

```
<HTML>

<HEAD>
<TITLE>Nombre de la página</TITLE>
</HEAD>

<BODY>
<H3>Primero una cabecera ...</H3>

<HR>

<APPLET CODE="HolaApplet.class" WIDTH=300 HEIGHT=60> </APPLET>

<HR>
</BODY>

</HTML>
```

Para comprobar el funcionamiento correcto, llama al navegador y abre el fichero `HolaApplet.html`.