Some tips about the Linux Shell

Speaker: Carlos Beltrán-González

October 15, 2007
Introduction

As you have noticed this year we will use Linux Ubuntu for the Operating Systems exercise. The main goals are:

1. Provide a very accessible way to use the computer (LiveCD). This allow you to follow the exercises everywhere you may have access to a computer.

2. Provide the essentials for the use of a Linux box

3. Practice how to program in a Linux box in C++

4. Practice some C++ programing techniques (classes, multithread, communications)

5. Learn how to use some basic functions of the YARP (Yet Another Robotic Platform) library
As you have noticed this year we will use Linux Ubuntu for the Operating Systems exercise. The main goals are:

1. Provide a very accessible way to use the computer (LiveCD). This allow you to follow the exercises everywhere you may have access to a computer.

2. Provide the essentials for the use of a Linux box

3. Practice how to program in a Linux box in C++

4. Practice some C++ programing techniques (classes, multithread, communications)

5. Learn how to use some basic functions of the YARP (Yet Another Robotic Platform) library
INTRODUCTION

As you have noticed this year we will use Linux Ubuntu for the Operating Systems exercise. The main goals are:

1. Provide a very accessible way to use the computer (LiveCD). This allow you to follow the exercises everywhere you may have access to a computer.
2. Provide the essentials for the use of a Linux box.
3. Practice how to program in a Linux box in C++
4. Practice some C++ programing techniques (classes, multithread, communications)
5. Learn how to use some basic functions of the YARP (Yet Another Robotic Platform) library

Speaker: Carlos Beltrán-González
As you have noticed this year we will use Linux Ubuntu for the Operating Systems exercise. The main goals are:

1. Provide a very accessible way to use the computer (LiveCD). This allow you to follow the exercises everywhere you may have access to a computer.

2. Provide the essentials for the use of a Linux box

3. Practice how to program in a Linux box in C++

4. Practice some C++ programing techniques (classes, multithread, communications)

5. Learn how to use some basic functions of the YARP (Yet Another Robotic Platform) library
Introduction

As you have noticed this year we will use Linux Ubuntu for the Operating Systems exercise. The main goals are:

1. Provide a very accessible way to use the computer (LiveCD). This allow you to follow the exercises everywhere you may have access to a computer.

2. Provide the essentials for the use of a Linux box

3. Practice how to program in a Linux box in C++

4. Practice some C++ programming techniques (classes, multithread, communications)

5. Learn how to use some basic functions of the YARP (Yet Another Robotic Platform) library
Some tips about the Linux Shell

Programming in a Linux Ubuntu box

why using Linux?

Main Goals

GNU Linux Environment

- Provide a very accessible way to use the computer (LiveCD). This allow you to follow the exercises everywhere you may have access to a computer.
- Provide the essentials for the use of a Linux box
- Practice how to program in a Linux box in C++

Esercitazioni

- Practice some C++ programming techniques (classes, multithread, communications)
- Learn how to use some basic functions of the YARP (Yet Another Robotic Platform) library
Main goals

**GNU Linux environment**
- Provide a very accessible way to use the computer (LiveCD). This allow you to follow the exercises everywhere you may have access to a computer.
- Provide the essentials for the use of a Linux box
- Practice how to program in a Linux box in C++

**Esercitazioni**
- Practice some C++ programming techniques (classes, multithread, communications)
- Learn how to use some basic functions of the YARP (Yet Another Robotic Platform) library
Some tips about the Linux Shell

Programming in a Linux Ubuntu box

Linux filesystem

A Typical Linux File System

/ directory root
the point where
all other directories
are mounted

/bin
all built-in
programs

/etc
config info for
The system

/home
user folders
And files

/mnt
mounted
file systems

/root
files & settings
for root

/sbin
built in progs
usually for root

/usr
third party
progs & extras

/var
mostly log data
& process info

/etc/init.d
start-up info
about which
services to run

/home/tom
folders & files
for the user
Tom

/home/jill
folders & files
for the user
Jill

/usr/bin
third party
programs

/usr/include
program code
for compiling
software

/usr/share
shared application
data - icons etc

/usr/src
uncompiled
source code

/var/run
PID progs
which avoid
multiple instances of
programs

Figure: The Linux’s filesystem structure

Speaker: Carlos Beltrán-González

Some tips about the Linux Shell
Some tips about the Linux Shell

- The linux’s Shell
- Some Shell’s commands

**COMMAND: man**

- Gives you a ”manual page” about a particular command
- Use: `man ls`
- Give you the man page of the `ls` command

**Advance use**

- `-al` Creates a list of ”hidden” files
- Use: `ls -al`
Some tips about the Linux Shell

- The linux’s Shell
- Some Shell’s commands

**COMMAND:**  *man*

- Gives you a ”manual page” about a particular command
- Use: `man ls`
- Give you the man page of the *ls* command

**ADVANCE USE**

- `-al` Creates a list of ”hidden” files
- Use: `ls -al`
Some tips about the Linux Shell

- The linux’s Shell
- Some Shell’s commands

**COMMAND:** *ls*

- Creates a list of the available files in the directory
- **Use:** *ls*

**Advance use**

- *-al* Creates a list of ”hidden” files
- **Use:** *ls -al*
**Command:** `ls`

- Creates a list of the available files in the directory

**Use:** `ls`

**Advance use**

- `-al` Creates a list of ”hidden” files

**Use:** `ls -al`
Some tips about the Linux Shell

- The Linux’s Shell
- Some Shell’s commands

**Command:** `mv`  
- Moves a file  
- Use: `mv filea.cpp fileb.cpp`

**Advance use**  
- It works also with directories  
- Use: `mv directorysource directorytarget`
COMMAND: *mv*

- Moves a file
- Use: `mv filea.cpp fileb.cpp`

ADVANCE USE

- It works also with directories
- Use: `mv directorysource directorytarget`
Some tips about the Linux Shell

- The linux’s Shell
- Some Shell’s commands

**COMMAND: cp**

- Copies (cp) a file
- Use: `cp filea.cpp fileb.cpp`

**Advance use**

- It works also with directories
- Use: `cp -R directorysource directorytarget`
- `-R` means "recursive"
Some tips about the Linux Shell

The Linux’s Shell

Some Shell’s commands

**COMMAND:** `cp`

- Copies (cp) a file
- Use: `cp filea.cpp fileb.cpp`

**ADVANCE USE**

- It works also with directories
- Use: `cp -R directorysource directorytarget`
- `-R` means ”recursive”
Some tips about the Linux Shell

The Linux's Shell

Some Shell’s commands

**COMMAND:** *rm*

- Removes files or directories
- **Use:** `rm file.cpp`

**Advance use**

- It works also with directories
- **Use:** `rm -rf directory`
- `-rf` makes the removal recursive and silent
Some tips about the Linux Shell

The Linux's Shell

Some Shell's commands

COMMAND:  `rm`

- Removes files or directories
- Use:  `rm file.cpp`

ADVANCE USE

- It works also with directories
- Use:  `rm -rf directory`
- `-rf` makes the removal recursive and silent
Some tips about the Linux Shell

- The Linux's Shell
- More complex use of the shell commands

MORE COMPLEX USE OF THE SHELL

STANDARD INPUT, STANDARD OUTPUT, STANDARD ERROR

- Everything is a file
- Devices can have a input stream, output stream...etc
- These data stream can be seen as "pipes" where data flows

REDIRECTIONS AND PIPES

- test
- standard output
- standard output
- A pipe is a flow connections represented by the symbol

EXAMPLES

- `ps aux | grep gnome`
Some tips about the Linux Shell

The Linux's Shell

More complex use of the shell commands

MORE COMPLEX USE OF THE SHELL

STANDARD INPUT, STANDARD OUTPUT, STANDARD ERROR

- Everything is a file
- Devices can have a input stream, output stream...etc
- These data stream can be seen as "pipes" where data flows

REDIRECTIONS AND PIPES

- test
- standard output
- standard output
- A pipe is a flow connections represented by the symbol

EXAMPLES

- ps aux — grep gnome

Speaker: Carlos Beltrán-González
MORE COMPLEX USE OF THE SHELL

STANDARD INPUT, STANDARD OUTPUT, STANDARD ERROR

- Everything is a file
- Devices can have a input stream, output stream...etc
- These data stream can be seen as "pipes" where data flows

REDIRECTIONS AND PIPES

- test
- standard output
- standard output
- A pipe is a flow connections represented by the symbol

EXAMPLES

- `ps aux | grep gnome`