An Introduction to Linux (Ubuntu) in Robotics

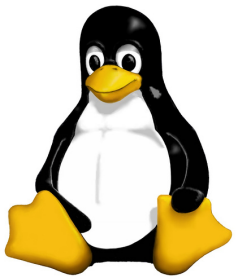
Juan M. Gandarias
jmgandarias@uma.es

Robotics and Mechatronics
Systems Engineering and Automation Department
University of Malaga

May 13, 2024

Overview

What is Linux?



- Family of open-source Unix-like operating systems based on the Linux kernel
- Packaged as a Linux distribution (distro), which includes the kernel and supporting system software and libraries
- A distro is an operating system made from a software collection that includes the Linux kernel, and often a package management system
- Created in 1991, by Linus Torvalds
- Set of utilities that will make your life much easier
- Linux Course
- Linux FileSystem

What is Ubuntu?

- Ubuntu is a Linux distribution based on Debian
- Composed (mostly) of free and open-source software
- Upgrade released every six months, with long-term support (LTS) releases every two years
- Ubuntu 18.04.6 LTS (Bionic Beaver)
- Ubuntu 20.04.6 LTS (Focal Fossa)
- Current version: Ubuntu 22.04.3 LTS (Jammy Jellyfish)

How to install Ubuntu (dual boot)

Here you can find some tutorials and videos on how to install Ubuntu on your PC

- RoboRescue UMA tutorial
- free code camp tutorial
- Youtube video

Ubuntu in Robotics

- Open-Source: Free, customizable and flexible
- Community support
- Ubuntu Robotics
- We "need" Ubuntu, because we want Robot Operating System (ROS) (among other things)
- User-friendly with git. Suggested git tools siute: gitkraken (It needs Github PRO $\rightarrow$ UMA students have it :))
- Customization and flexibility
- Real-Time capabilities (RT kernel)
- Resource efficiency
  - Suitable for embedded systems with limited hardware resources
  - Minimal system requirements allow for running on a variety of hardware platforms

How to work with the terminal

- The terminal is a powerful tool for efficient and precise control of a Linux system
- Regular practice and exploration are key to mastering the command line interface
- Accessing the Terminal:
  Use the keyboard shortcut $Ctrl + Alt + T$ or search for "Terminal" in the application menu
- Suggestion (useful to work with robots) $\rightarrow$ Terminator

Basic Commands

- `pwd`: Print current working directory
- `ls`: List files and directories
- `cd`: Change directory
- `mkdir`: Create a new directory
- `touch`: Create a new empty file
- `cp`: Copy files or directories
- `mv`: Move or rename files/directories
- `rm`: Remove files or directories

File System Navigation & Manipulation

- Use `cd` to navigate through directories
- Use `cd..` to go up one level
- Use absolute or relative paths for navigation
- Copy files: `cp source destination`
- Move files: `mv source destination`
- Remove files: `rm filename`
- Create a symbolic link: `ln -s source link_name`

Text Manipulation

- `cat`: Display the content of a file
- `nano` or `vim`: Text editors for creating or editing files
- `grep`: Search for specific patterns in files
- `echo`: Print text to the terminal

Permissions

- `chmod`: Change file permissions
- `chown`: Change file ownership
- `chgrp`: Change file group

1. **Introduction to Bash Scripting**
   - Bash is a command processor that typically runs in a text window
   - Bash scripting involves writing sequences of commands to automate tasks
   - Convention
2. **Script Execution**
   - Scripts are text files with a series of commands.
   - Execute a script by making it executable (`chmod +x script.sh`) and running it (`./script.sh`).
3. **Variables**
   - Variables store data for later use.
   - Example: `name="John"`.

Bash Scripting (cont.)

4. **Conditions and Loops**
   - Use `if`, `else`, and `fi` for conditional statements.
   - Use `for` and `while` loops for repeated tasks.
5. **Example Bash Scripts**

- **Script 1: Hello World**

```
#!/bin/bash
echo ‘‘Hello, World!’’
```

- **Script 2: User Greeting**

```
#!/bin/bash
echo ‘‘Enter your name:’’
read NAME
echo ‘‘Hello, $NAME!’’
```

- **Script 3: File Count**

```
#!/bin/bash
echo ‘‘Number of files in
the current directory:’’
ls | wc -l
```

  - | redirects the output of one command as the input to another one
  - *wc* (word count). -l refers tothe number of lines in the input it receives.

Bash Scripting (cont.)

- **Script 4: User Files Count**

```bash
#!/bin/bash

# Ask for the name
echo "Please enter your name:"
read name

# Check if the name is "Juanma"
if [ "$name" == "Juanma" ]; then
    # Get the number of files and the list of files
    num_files=$(ls -l | grep "^-" | wc -l)
    files_list=$(ls)

    # Display the number of files
    echo "Number of files in the folder: $num_files"

    # Display the list of files
    echo "List of files:"
    echo "$files_list"
else
    # If the name is not "Juanma", display unauthorized message
    echo "Sorry, you are not an authorized user."
fi
```

Aliases

- Shortcuts or custom commands to simplify and abbreviate longer commands
- Useful for creating personalized shortcuts for frequently used commands
- `alias`: show your current aliases
- `alias alias_name = 'command'`: create the alias `alias_name` that runs `command`.

  `echo ''alias_name = 'command' '' >> ~/.bashrc source ~/.bashrc`: Make aliases presistent across sessions (add to your shell configuration file)

  E.g., `alias file_count='cd ~/examples/; ./file_count.sh'`