Basic Tools for NLP

Rishu Kumar, Iuliia Zaitova, Chris Hyek

Saarland University

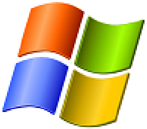
27.04.2023

Class Outline

- Introducing Unix Principles
- Navigating Files & Directories
- Managing Users
- Managing Processes and Services

Why Unix?

Microsoft's Windows NT-based OS



Everything else that traces back to Unix

Linux, Mac OS X, Android, iOS, Chrome OS → *Nix systems

Minor Notes

- If you use Mac, your default shell is **zsh** instead of **bash**[1]
- While most of the basic commands will work, some binaries might not
- If you have Ubuntu, you might have *tree* installed on your machine, if you have Mac, look at Brew at brew.sh (install it now)

[1] Some Licensing issues with the latest version of Bash

Minor Notes

- If you use Mac, your default shell is **zsh** instead of **bash**[2]
- While most of the basic commands will work, some binaries might not
- If you have Ubuntu, you might have *tree* installed on your machine, if you have Mac, look at Brew at brew.sh (install it now)
- Do *brew install tree* once you are done installing it
- While, it installs, we look at a meme!

[2] Some Licensing issues with the latest version of Bash

UNIVERSITÄT
DES
SAARLANDES

Obligatory Microsoft Meme

Unix’s Design Principles

- Each program does one thing well
- Programs work together to create a system (pipelining)
- The importance of textual data
- “Everything is a file” → “Everything appears somewhere in the file system” *(That’s why only Windows has drive letters)*

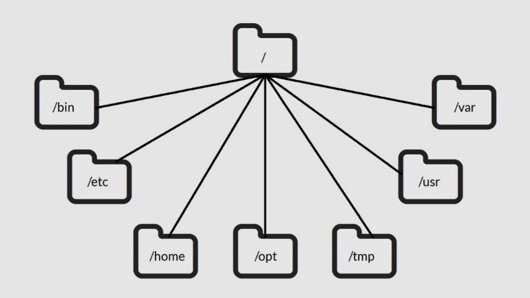
Unix File System – methodology for logically analyzing and storing data such that the system is easy to manage

Two main components:

File *(always placed under directory)*
Contain information/data

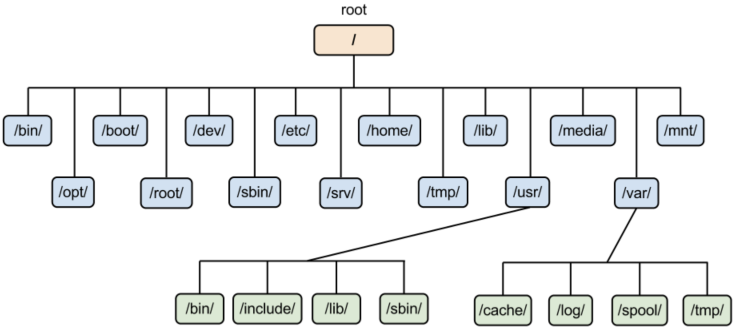
Directory *(special file that contains other files/dir-s)*

top node is called the **root node**



Have a look at your file system!

1. Go to root directory `cd /`
2. List dir-s/files in the current directory `ls` `tree -L 1`
3. Do you see different files?



Some directories

- /bin: short for binaries, this is the directory where many commonly used executable commands reside
- /dev: contains device-specific files
- /etc: contains system configuration files
- /lib: contains all library files
- /proc: contains files related to system processes
- /root: the root users' home directory (different than ' / ')
- /tmp: storage for temporary files that are periodically removed from the file system
- /var: It is a short form for 'variable', a place for files that may often change

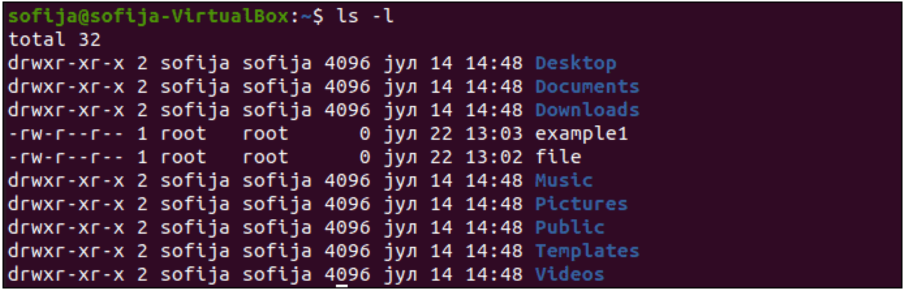
Hidden files

1. Go to home directory `cd`
2. List hidden files
   `ls -a`

You can now see files that start with a "." symbol (*.profile, .ssh, .bashrc, .zshrc*)
But we would leave them for later :)

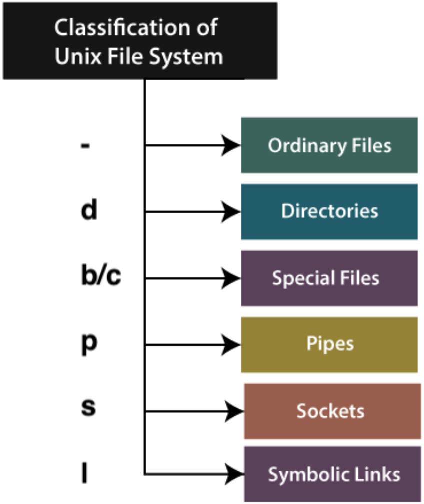
Types of files in Unix

1. Get more information about listed files ls -l
2. See something like this:



```
sofija@sofija-VirtualBox:~$ ls -l
total 32
drwxr-xr-x 2 sofija sofija 4096 јул 14 14:48 Desktop
drwxr-xr-x 2 sofija sofija 4096 јул 14 14:48 Documents
drwxr-xr-x 2 sofija sofija 4096 јул 14 14:48 Downloads
-rw-r--r-- 1 root   root      0 јул 22 13:03 example1
-rw-r--r-- 1 root   root      0 јул 22 13:02 file
drwxr-xr-x 2 sofija sofija 4096 јул 14 14:48 Music
drwxr-xr-x 2 sofija sofija 4096 јул 14 14:48 Pictures
drwxr-xr-x 2 sofija sofija 4096 јул 14 14:48 Public
drwxr-xr-x 2 sofija sofija 4096 јул 14 14:48 Templates
drwxr-xr-x 2 sofija sofija 4096 јул 14 14:48 Videos
```

File Types



Classification of Unix File System

- → Ordinary Files

d → Directories

b/c → Special Files

p → Pipes

s → Sockets

l → Symbolic Links

More File Types

- **Special Files**: Files that represent hardware devices (e.g. /dev/printer, /dev/sda) or system resources (e.g. /dev/null, /proc/cpuinfo)
- **Pipes**: A mechanism for inter-process communication that allows the output of one process to be used as the input of another process (e.g. ls | grep file)
- **Sockets**: A type of file used for inter-process communication between processes on the same or different computers (e.g. for a web browser connecting to a web server using a TCP socket).
- **Symbolic Links**: A file that acts as a pointer to another file or directory, allowing for easier navigation of the file system (e.g. /usr/local/bin/python -> /usr/bin/python3.9)

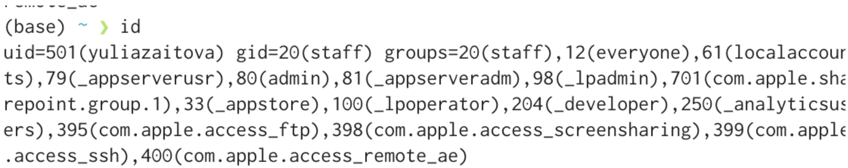
Managing users/groups
Also file-based. Two most important files:

- **/etc/passwd**: Contains user account information, including the user's username, user ID (UID), home directory, and default shell.
- **/etc/group**: Contains group account information, including the group's name, group ID (GID), and a list of members.

Root user / superuser — has complete control over the system and can perform any action without restrictions. (use the root user account with caution!!)

Some commands to manage your user account:

- passwd: Used to change yours / smb's password
- whoami: Displays the username of the current user
- id: Displays user id, username, and groups that the user belongs to



```
(base) ~ ❯ id
uid=501(yuliazaitova) gid=20(staff) groups=20(staff),12(everyone),61(localaccour
ts),79(_appserverusr),80(admin),81(_appserveradm),98(_lpadmin),701(com.apple.sha
repoint.group.1),33(_appstore),100(_lpoperator),204(_developer),250(_analyticsus
ers),395(com.apple.access_ftp),398(com.apple.access_screensharing),399(com.apple
.access_ssh),400(com.apple.access_remote_ae)
```

Group memberships determine the user's access to files and directories that have been assigned specific permissions for certain groups.

- **chmod**: Changes the permissions of files or dir

The `chmod` command is used to change the permissions of files or directories. It can be used to grant or revoke **read, write, and execute** permissions for users and groups.

Syntax:

```
chmod [options] mode file(s)
```

Options:

- `-R`: Change the permissions recursively for all files and directories in the specified directory.
- `-v`: Show a message for each file or directory that is modified.

Mode: The mode argument specifies the new permissions.

- r (read): 4
- w (write): 2
- x (execute): 1

`chmod 644 file.txt` - grants read and write permissions to the owner, and read-only permissions to the group and others.

Exercise Time!

1. Create a new directory called `chmod-exercise` (`mkdir chmod-exercise`).
2. Do chmod 400 chmod-exercise
3. Create a new file called `secret.sh` inside the `chmod-exercise` directory (`touch chmod-exercise/secret.sh`).
4. Change the permission settings again

Exercise Time!

1. Create a new file called `secret.sh` inside the `chmod-exercise` directory (`touch chmod-exercise/secret.sh`).
2. Set the permissions of `secret.sh` so that only the owner can read and write to it.

   ```
   chmod 600 chmod-exercise/secret.sh
   ```

3. Use the `ls -l` command to verify the permissions of `secret.sh`.

   ```
   ls -l chmod-exercise/secret.sh
   ```

4. Change the permissions of `secret.sh` so that the owner can read and write to it, and the group and others can only read it.

   ```
   chmod 644 chmod-exercise/secret.sh
   ```

5. Use the `ls -l` command to verify the new permissions of `secret.sh`.

   ```
   ls -l chmod-exercise/secret.sh
   ```

6. Delete the `chmod-exercise` directory and its contents.

   ```
   rm -r chmod-exercise
   ```