

# Linux Tutorials Outline

## Today:

- Introduction to Linux, basic apps and utilities (file transfer, text editors, file managers, etc.)

## 19 January (next Saturday):

- Standard Input/Output in Linux, pipes, redirection, regular expressions, text utils (sed, grep, etc.)
- Processes, signals, IPC in Linux
- **Assignment 1**

## 2 February:

- SSH: tunneling, reverse SSH, public key authentication, etc.
- Basic scripting

## 16 February:

- Virtualization and Docker
- **Assignment 2**

# Introduction to Linux

History of Linux, basic concepts, file system

Nikita Neveditsin, SMU, 2019

[nikita.neveditsin@smu.ca](mailto:nikita.neveditsin@smu.ca)



# Why learn Linux?

- Free, fast and robust
- Runs on the most diverse hardware (servers, smartphones, single-board computers, routers, etc.)
- Great for programming
- Variety of tools for working with text data
- Best choice for research purposes (powerful servers usually use Linux)
- Nice to have it on your resume



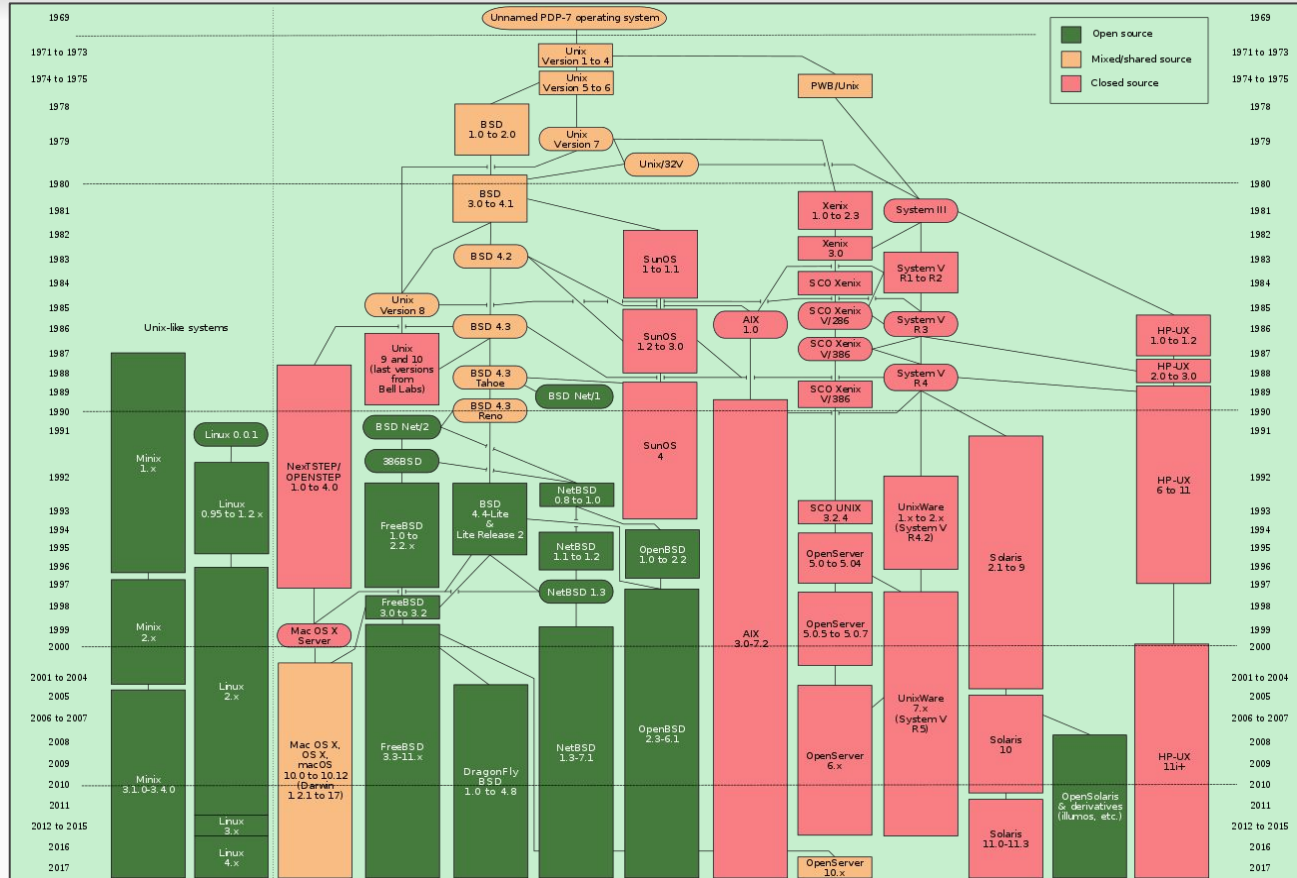
# Brief history of Linux

- Linux is a family of free and open-source software operating systems built around the Linux kernel (first released on September 17, 1991, by Linus Torvalds)[\[1\]](#)
- Linux is a UNIX-like operating system
- UNIX was developed in 1969 at Bell Labs research center by Ken Thompson, Dennis Ritchie, and others
- Linux is mostly POSIX-compliant OS (Portable Operating System Interface is a family of standards specified by the IEEE for maintaining compatibility between operating systems)
- All UNIX (and UNIX-like) operating systems follow these simplified principles (UNIX philosophy) [\[2\]](#):
  - Write programs that do one thing and do it well
  - Write programs to work together (IPC, pipes, redirection)
  - Write programs to handle text streams, because that is a universal interface.

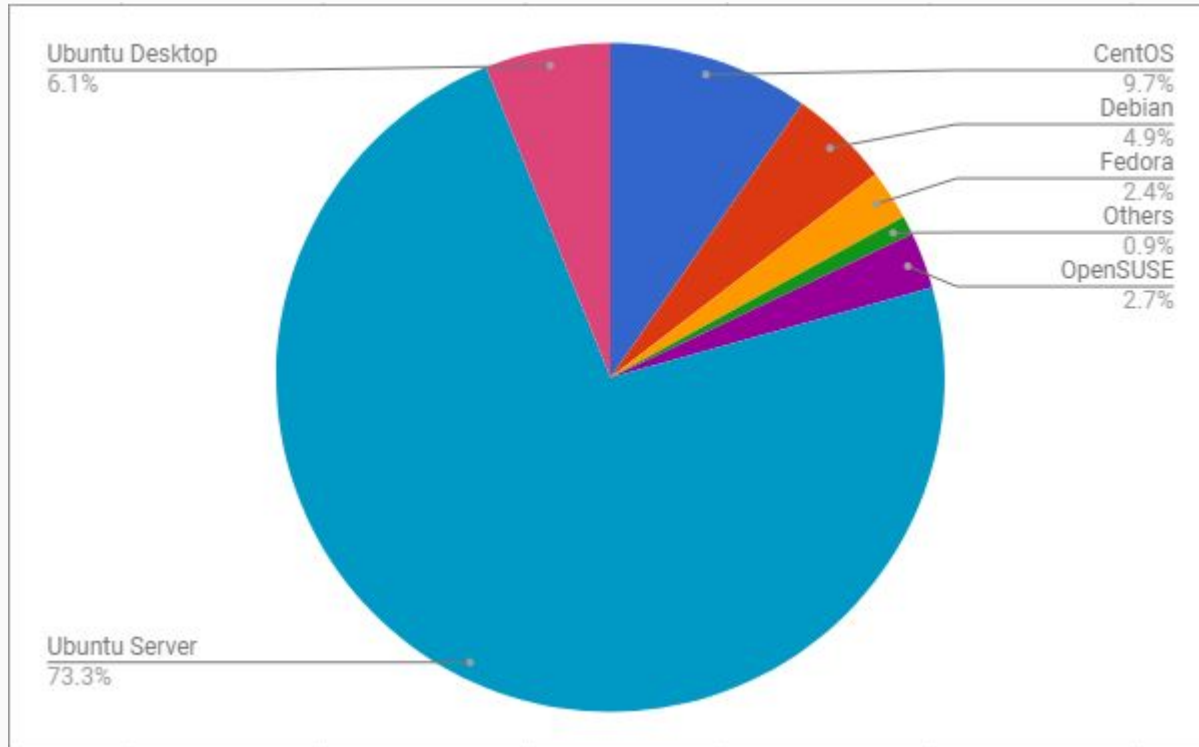


[\[link\]](#)

# Evolution of UNIX [3]



# Most popular Linux distributions [\[4\]](#)



# Linux vs Windows

	<b>Linux</b>	<b>Windows</b>
Licence	Free	Proprietary
Code	Open source	Closed Source
How to use	Mostly command line	GUI
What is it good for?	Text processing, hosting and development of server applications, research	Office apps, Internet browsing, multimedia, games

# Basics of Linux OS. Filesystem hierarchy

- Windows uses backslash (C:\test\program.exe), Linux uses forward slash (/bin/ls)
- Windows uses drive letters (C:\ D:\), Linux uses virtual file system with MOUNT POINTS (tree)
- Names in Linux are CAsE - Sensitive!



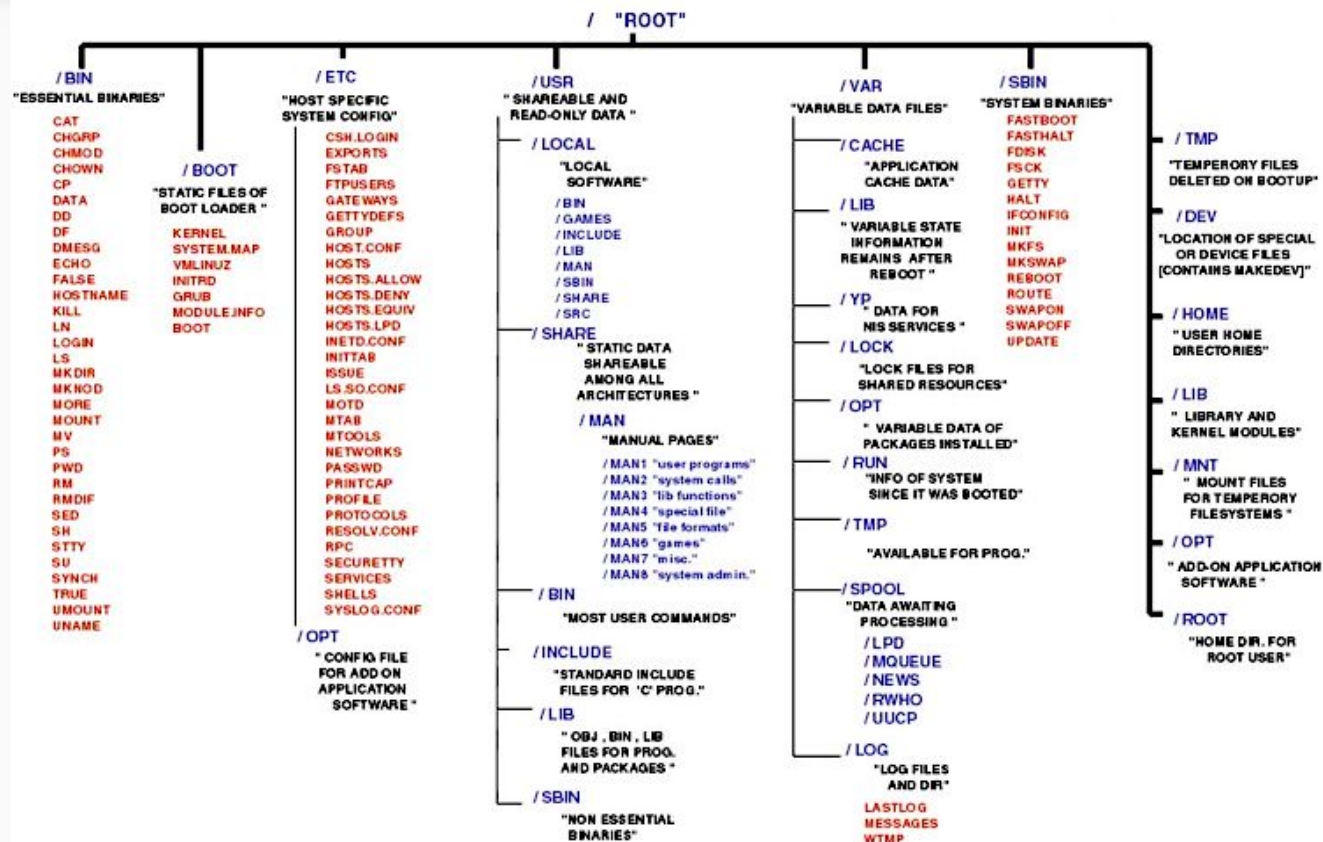
[link](#)

Pathnames: **ABSOLUTE** (/home/user1/file1.txt): full path from the root and **RELATIVE** (file1.txt if you are in the directory where file1.txt located)

Special symbols: . (dot) - is a current directory, .. (double dot) is a parent directory, ~ (tilde) is home directory



# Basics of Linux OS. Filesystem hierarchy [6]



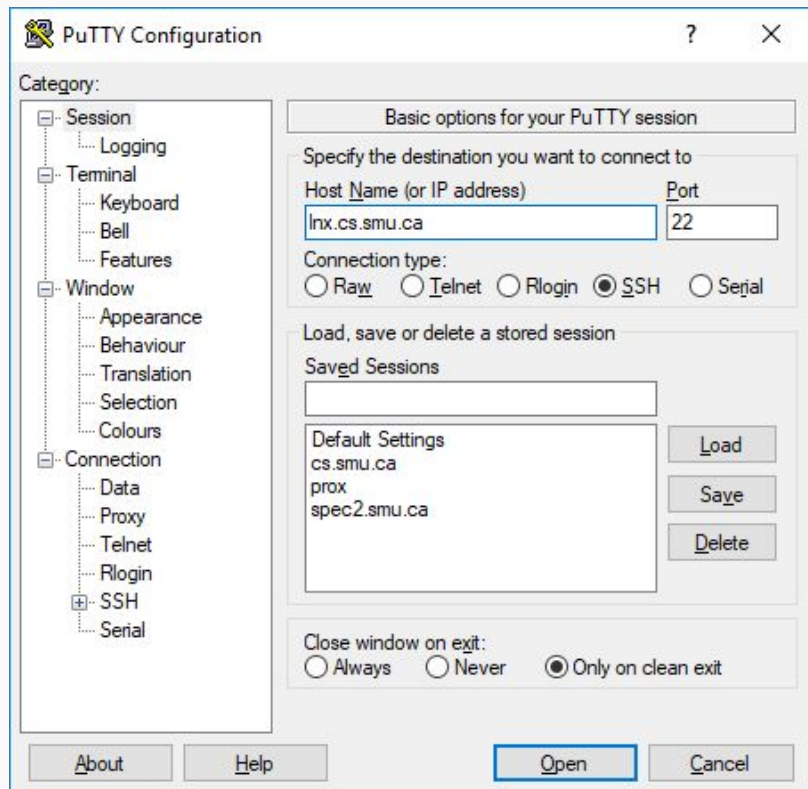
# Connecting to remote Linux Machine via SSH

- Windows: PuTTY
- Linux: ssh from terminal
- MAC: ssh from terminal

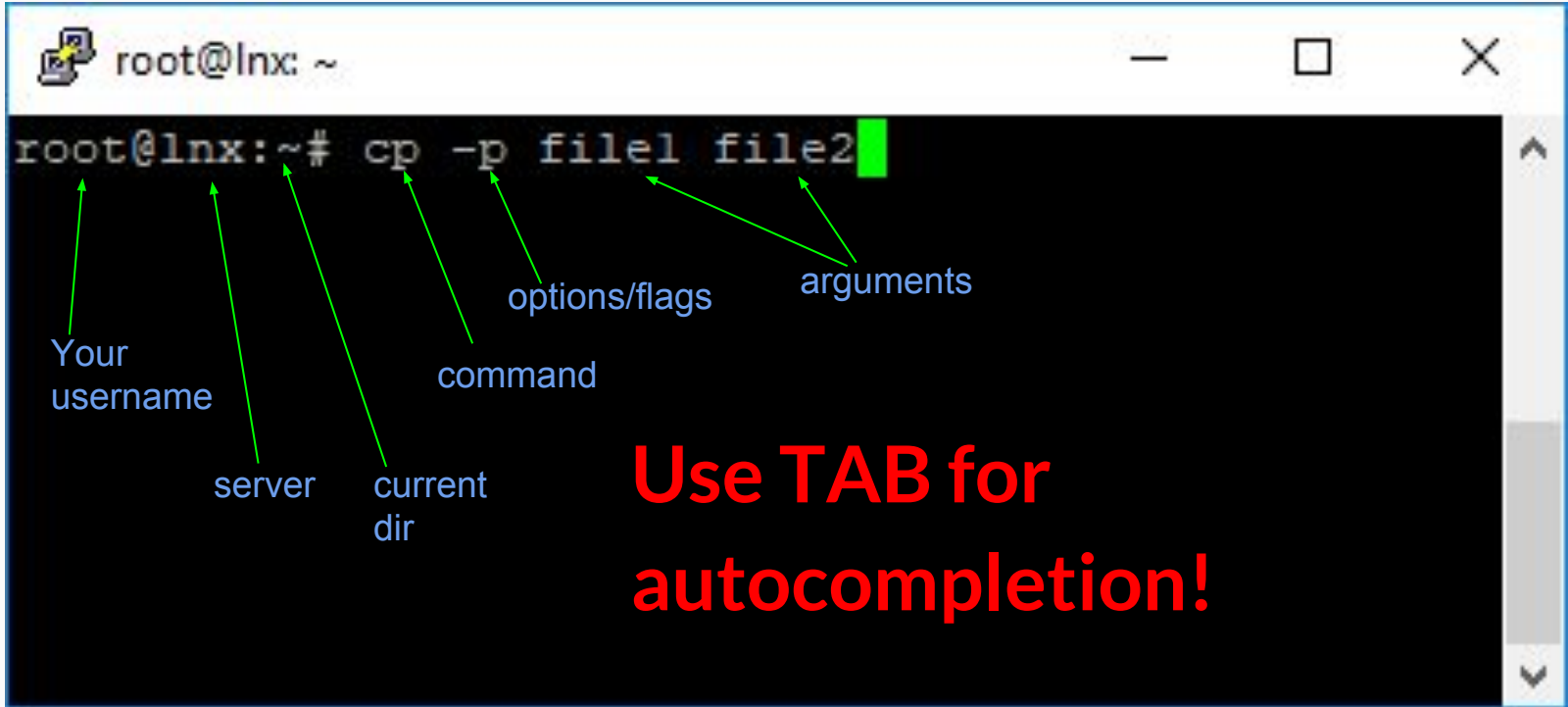
Address: lnx.cs.smu.ca

Username: from the list

Password: A#



# Linux shell interface



A screenshot of a Linux terminal window. The window title bar shows a terminal icon, the text "root@lnx: ~", and standard window controls (minimize, maximize, close). The terminal content shows the prompt "root@lnx:~#" followed by the command "cp -p file1 file2" with a green cursor at the end. Green arrows point from labels to parts of the command: "Your username" points to "root", "server" points to "lnx", "current dir" points to "~", "command" points to "cp", "options/flags" points to "-p", and "arguments" points to "file1 file2".

```
root@lnx:~# cp -p file1 file2
```

Labels and arrows:

- Your username (points to root)
- server (points to lnx)
- current dir (points to ~)
- command (points to cp)
- options/flags (points to -p)
- arguments (points to file1 file2)

**Use TAB for  
autocompletion!**

# Basic commands. Where am I?

**pwd** - print name of current/working directory (gives ABSOLUTE path)

---

 root@lnx: /home/student

```
root@lnx:~# pwd
/root
root@lnx:~# cd /home/student/
root@lnx:/home/student# pwd
/home/student
root@lnx:/home/student#
```

# Basic commands. What's in the directory?

**ls** - list directory contents

Useful flags:

- **-R** - recursively;
- **-l** - as listing;
- **-h** - human-readable size
- **-a** - show hidden (starting with dot . files)
- **-S** sort by size

The image shows a terminal window with the command `ls -lha` executed in the `/etc/ssh` directory. The output is a long listing of files and directories. Annotations with green arrows point to specific parts of the output:

- permissions**: Points to the first column of permissions (e.g., `drwxr-xr-x`).
- owner and group**: Points to the second and third columns (e.g., `2 root root`).
- Number of links**: Points to the fourth column (e.g., `1`).
- size**: Points to the fifth column (e.g., `4.0K`).
- Last modification date**: Points to the sixth and seventh columns (e.g., `Jan 23 06:43`).

The terminal output is as follows:

```
cssmuadm@lnx:/etc/ssh$ ls -lha
total 348K
drwxr-xr-x  2 root root 4.0K Jan 23 06:43 .
drwxr-xr-x 113 root root 4.0K May  1 19:28 ..
-rw-r--r--  1 root root 294K Aug 11 2016 moduli
-rw-r--r--  1 root root 1.8K Aug 11 2016 ssh_config
-rw-r--r--  1 root root 2.5K Apr 21 2017 sshd_config
-rw-----  1 root root 668 Apr 21 2017 ssh_host_dsa_key
-rw-r--r--  1 root root 601 Apr 21 2017 ssh_host_dsa_key.pub
-rw-----  1 root root 227 Apr 21 2017 ssh_host_ecdsa_key
-rw-r--r--  1 root root 173 Apr 21 2017 ssh_host_ecdsa_key.pub
-rw-----  1 root root 399 Apr 21 2017 ssh_host_ed25519_key
-rw-r--r--  1 root root 93 Apr 21 2017 ssh_host_ed25519_key.pub
-rw-----  1 root root 1.7K Apr 21 2017 ssh_host_rsa_key
-rw-r--r--  1 root root 393 Apr 21 2017 ssh_host_rsa_key.pub
-rw-r--r--  1 root root 338 Jan  6 2017 ssh_import_id
```

# Basic commands. Go to other directory

## cd - change directory

```
root@lnx: /  
root@lnx:~# cd /var/1  
lib/  local/ lock/  log/  
root@lnx:~# cd /var/log/  
apache2/      dbconfig-common/  fsck/      mysql/      tomcat8/  
apt/          dist-upgrade/     lxd/       nginx/      unattended-upgrades/  
root@lnx:~# cd /var/log/  
apache2/      dbconfig-common/  fsck/      mysql/      tomcat8/  
apt/          dist-upgrade/     lxd/       nginx/      unattended-upgrades/  
root@lnx:~# cd /var/log/fsck/  
root@lnx:/var/log/fsck# cd ../../..  
root@lnx:/# ls  
bin  dev  home  initrd.img.old  lib64  media  opt  root  sbin  srv  tmp  var  vmlinuz.old  
boot  etc  initrd.img  lib  lost+found  mnt  proc  run  snap  sys  usr  vmlinuz  
root@lnx:/# pwd  
/  
root@lnx:/#
```

# Basic commands. Create a new directory

**mkdir** - make directories

Useful flags:

- **-p** - make parent directories as needed

```
root@lnx: ~  
root@lnx:~# mkdir dir1  
root@lnx:~# mkdir dir2/dir3  
mkdir: cannot create directory 'dir2/dir3': No such file or directory  
root@lnx:~# mkdir -p dir2/dir3  
root@lnx:~# ls -R dir2  
dir2:  
dir3  
  
dir2/dir3:  
root@lnx:~#
```

# Basic commands. Delete files/dirs

**rm** - remove files or directories

Useful flags:

- **-f, --force** - ignore nonexistent files and arguments, never prompt
- **-i** - prompt before every removal
- **-r, -R, --recursive** - remove directories and their contents recursively

```
cssmuadm@lnx: ~  
cssmuadm@lnx:~$ ls  
dir1 file1  
cssmuadm@lnx:~$ rm file1  
cssmuadm@lnx:~$ rm dir1  
rm: cannot remove 'dir1': Is a directory  
cssmuadm@lnx:~$ rm -r dir1  
cssmuadm@lnx:~$ ls  
cssmuadm@lnx:~$
```



# rm: don't try to make your computer faster

## FILE COMMANDS

ls - directory listing

ls -al - formatted listing with hidden files

cd dir - change directory to dir

cd - change to home

pwd - show current directory

mkdir dir - create directory dir

rm file - delete file

rm -r dir - delete directory dir

rm -f file - force remove file

rm -rf dir - remove directory dir

rm -rf / - make computer faster

cp file1 file2 - copy file1 to file2

mv file1 file2 - rename file1 to file2

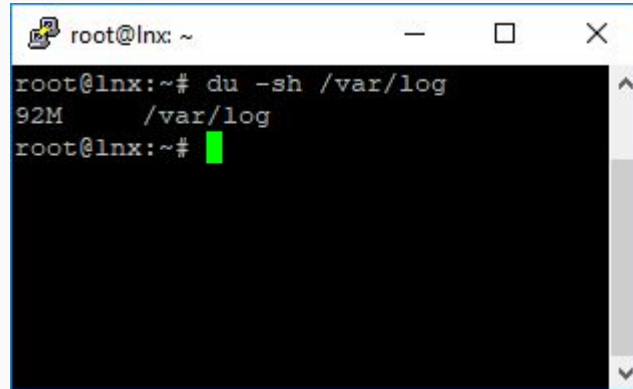
[\[link\]](#)

# Basic commands. What's file/dir size?

**du** - estimate file space usage

Useful flags:

- **-h** - human-readable size
- **-s** - total size of folder and its contents
- **--max-depth=1** or other number - how deep should we scan?



```
root@lnx: ~  
root@lnx:~# du -sh /var/log  
92M    /var/log  
root@lnx:~#
```

# Basic commands. Let's copy!

**cp** - copy files and directories

Useful flags:

- **-R, -r, --recursive**  
copy directories recursively
- **-p** - same as  
**--preserve=mode,ownership,**  
**timestamps**

```
cssmuadm@lnx: ~  
cssmuadm@lnx:~$ ls  
dir1 file1  
cssmuadm@lnx:~$ cp file1 file2  
cssmuadm@lnx:~$ cp dir1 dir2  
cp: omitting directory 'dir1'  
cssmuadm@lnx:~$ cp -r dir1 dir2  
cssmuadm@lnx:~$ ls  
dir1 dir2 file1 file2  
cssmuadm@lnx:~$
```

Source

Destination

# Basic commands. Move or rename

**mv** - move (rename) files

Just renames file1 to file2

Moves file2 to dir1

Moves dir1 to dir2

```
cssmuadm@lnx: ~  
cssmuadm@lnx:~$ ls -l  
total 8  
drwxrwxr-x 2 cssmuadm cssmuadm 4096 May  2 15:02 dir1  
drwxrwxr-x 2 cssmuadm cssmuadm 4096 May  2 15:02 dir2  
-rw-rw-r-- 1 cssmuadm cssmuadm  0 May  2 15:02 file1  
-rw-rw-r-- 1 cssmuadm cssmuadm  0 May  2 15:02 file2  
cssmuadm@lnx:~$ mv file1 file11  
cssmuadm@lnx:~$ ls  
dir1 dir2 file11 file2  
cssmuadm@lnx:~$ mv file2 dir1  
cssmuadm@lnx:~$ ls  
dir1 dir2 file11  
cssmuadm@lnx:~$ ls -R  
.:  
dir1 dir2 file11  
  
./dir1:  
file2  
  
./dir2:  
cssmuadm@lnx:~$ mv dir1 dir2  
cssmuadm@lnx:~$ ls -R  
.:  
dir2 file11  
  
./dir2:  
dir1  
  
./dir2/dir1:  
file2  
cssmuadm@lnx:~$
```

# Basic commands. Find or locate?

## locate - find files by name

- Pros: FASTER than find as uses a database which is updated daily by default (updatedb) and by some other applications (like app manager)
- Cons: some information may be outdated, does not have much options

```
root@lnx:~# locate gzip
/bin/gzip
/usr/lib/apt/methods/gzip
/usr/lib/klibc/bin/gzip
/usr/lib/python2.7/gzip.py
/usr/lib/python2.7/gzip.pyc
/usr/lib/python3.5/gzip.py
/usr/lib/python3.5/_pycache_/gzip.cpython-35.pyc
/usr/share/bash-completion/completions/gzip
/usr/share/doc/gzip
/usr/share/doc/gzip/README-release
/usr/share/doc/gzip/README.gz
/usr/share/doc/gzip/TODOLinux
/usr/share/doc/gzip/changelog.Debian.gz
/usr/share/doc/gzip/copyright
/usr/share/info/gzip.info.gz
/usr/share/man/man1/gzip.1.gz
/usr/share/mime/application/gzip.xml
/usr/share/vim/vim74/autoload/gzip.vim
/usr/share/vim/vim74/doc/pi_gzip.txt
/usr/share/vim/vim74/plugin/gzip.vim
/usr/src/linux-headers-4.4.0-109-generic/include/config/decompress/gzip.h
/usr/src/linux-headers-4.4.0-109-generic/include/config/have/kernel/gzip.h
/usr/src/linux-headers-4.4.0-109-generic/include/config/kernel/gzip.h
/usr/src/linux-headers-4.4.0-109-generic/include/config/rd/gzip.h
/usr/src/linux-headers-4.4.0-112-generic/include/config/decompress/gzip.h
/usr/src/linux-headers-4.4.0-112-generic/include/config/have/kernel/gzip.h
/usr/src/linux-headers-4.4.0-112-generic/include/config/kernel/gzip.h
/usr/src/linux-headers-4.4.0-112-generic/include/config/rd/gzip.h
/usr/src/linux-headers-4.4.0-116-generic/include/config/decompress/gzip.h
/usr/src/linux-headers-4.4.0-116-generic/include/config/have/kernel/gzip.h
/usr/src/linux-headers-4.4.0-116-generic/include/config/kernel/gzip.h
/usr/src/linux-headers-4.4.0-116-generic/include/config/rd/gzip.h
/usr/src/linux-headers-4.4.0-119-generic/include/config/decompress/gzip.h
/usr/src/linux-headers-4.4.0-119-generic/include/config/have/kernel/gzip.h
/usr/src/linux-headers-4.4.0-119-generic/include/config/kernel/gzip.h
/usr/src/linux-headers-4.4.0-119-generic/include/config/rd/gzip.h
/usr/src/linux-headers-4.4.0-121-generic/include/config/decompress/gzip.h
/usr/src/linux-headers-4.4.0-121-generic/include/config/have/kernel/gzip.h
/usr/src/linux-headers-4.4.0-121-generic/include/config/kernel/gzip.h
/usr/src/linux-headers-4.4.0-121-generic/include/config/rd/gzip.h
/var/lib/dpkg/info/gzip.list
/var/lib/dpkg/info/gzip.md5sums
root@lnx:~#
```

# Basic commands. Find or locate?

**find** - search for files in a directory hierarchy

- **-type** - usually **d** or **f** (dir or file)
- **-name** name of the file (can use wildcard \*)
- **-maxdepth**
- Can combine multiple conditions

```
root@lnx: ~  
root@lnx:~# find . -type f -name *history  
./bash_history  
./nano/search_history  
./node_repl_history  
./mysql_history  
root@lnx:~# find /bin -type f -name bash  
/bin/bash  
root@lnx:~# find /var/log -type d  
/var/log  
/var/log/nginx  
/var/log/dist-upgrade  
/var/log/unattended-upgrades  
/var/log/fsck  
/var/log/apt  
/var/log/tomcat8  
/var/log/mysql  
/var/log/apache2  
/var/log/lxd  
/var/log/dbconfig-common  
root@lnx:~# find ~ -maxdepth 1  
/root  
/root/.profile  
/root/.bash_history  
/root/.bashrc  
/root/.nano  
/root/test_cpy.sh  
/root/file1  
/root/.cache  
/root/dirl  
/root/.node_repl_history  
/root/.npm  
/root/test.sh  
/root/.ssh  
/root/.mysql_history  
root@lnx:~#
```

Starting dir

# Basic commands. Ask that man first

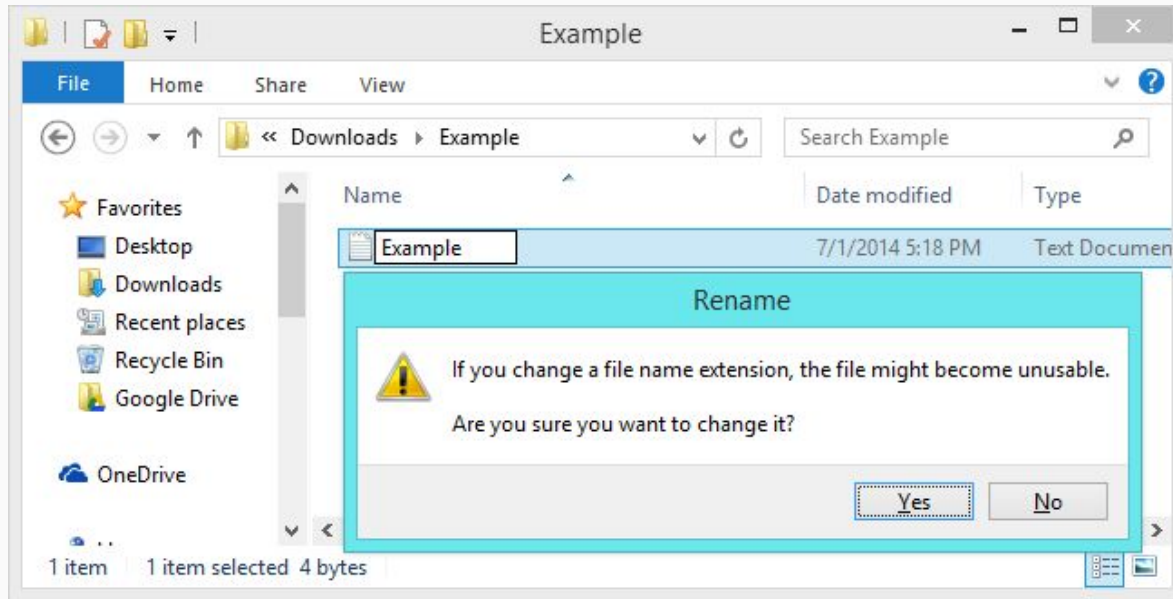
**man** - manual page for a command (e.g., man **killall**)

```
root@lnx: ~  
root@lnx:~# man killall
```

```
root@lnx: ~  
KILLALL(1)                                User Commands                                KILLALL(1)  
  
NAME  
    killall - kill processes by name  
  
SYNOPSIS  
    killall [-Z, --context pattern] [-e, --exact]  
            [-g, --process-group] [-i, --interactive]  
            [-o, --older-than TIME] [-q, --quiet] [-r, --regexp]  
            [-s, --signal SIGNAL, -SIGNAL] [-u, --user user]  
            [-v, --verbose] [-w, --wait] [-y, --younger-than TIME]  
            [-I, --ignore-case] [-V, --version] [--] name ...  
    killall -l  
    killall -V, --version  
  
DESCRIPTION  
    killall sends a signal to all processes running any of  
Manual page killall(1) line 1 (press h for help or q to quit)
```

information  
but

# Linux file types (regular files)

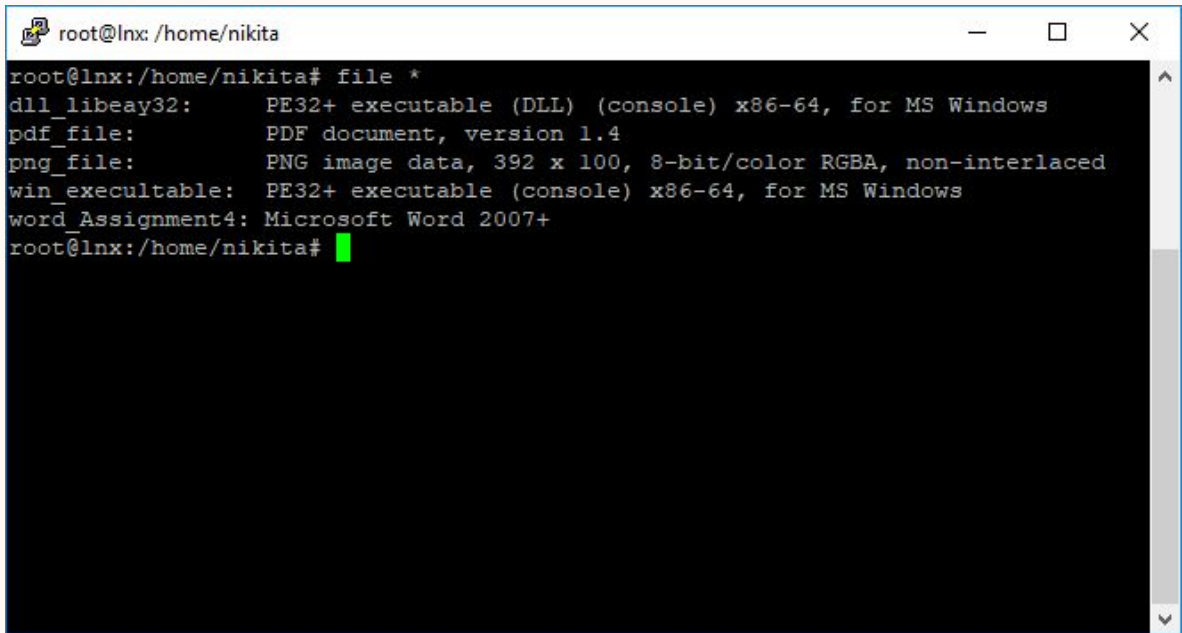


[\[link\]](#)



# Linux file types (regular files)

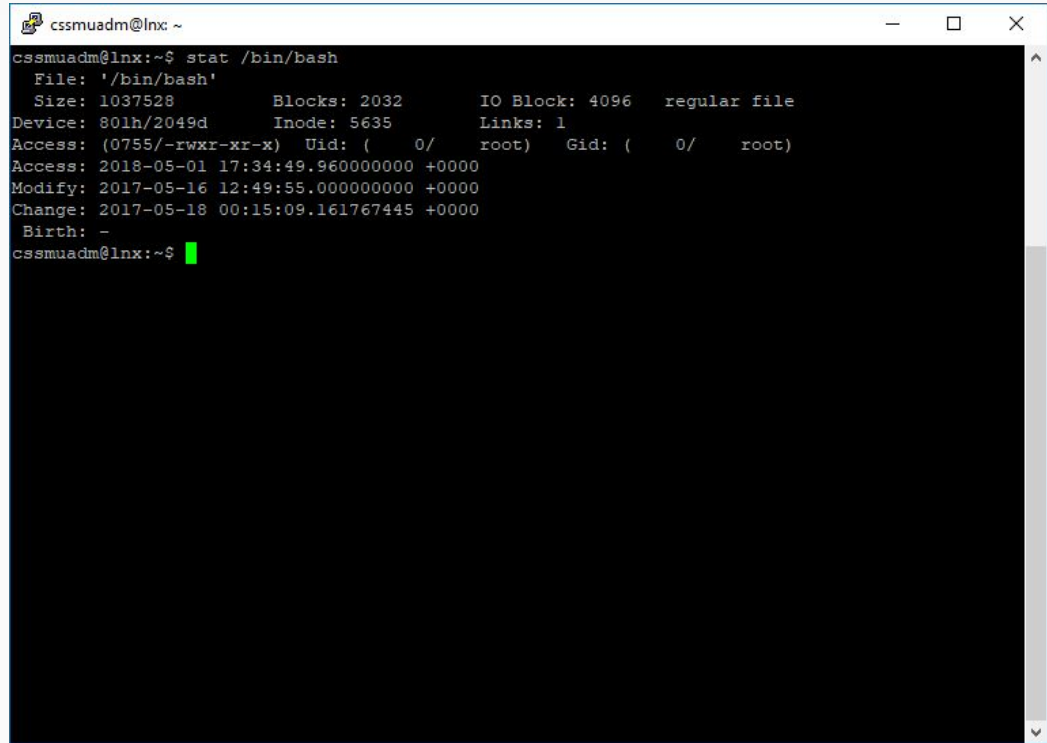
**file** -- determine file type

A terminal window titled 'root@lnx: /home/nikita' with standard window controls. The terminal shows the command 'file \*' and its output for several files. The output is as follows:

```
root@lnx:/home/nikita# file *
dll_libeay32:      PE32+ executable (DLL) (console) x86-64, for MS Windows
pdf_file:         PDF document, version 1.4
png_file:         PNG image data, 392 x 100, 8-bit/color RGBA, non-interlaced
win_executable:   PE32+ executable (console) x86-64, for MS Windows
word_Assignment4: Microsoft Word 2007+
root@lnx:/home/nikita#
```

# More on files

**stat** - display file (or file system) status



```
cssmuadm@lnx: ~  
cssmuadm@lnx:~$ stat /bin/bash  
  File: '/bin/bash'  
  Size: 1037528      Blocks: 2032      IO Block: 4096   regular file  
Device: 801h/2049d  Inode: 5635       Links: 1  
Access: (0755/-rwxr-xr-x)  Uid: (  0/   root)   Gid: (  0/   root)  
Access: 2018-05-01 17:34:49.960000000 +0000  
Modify: 2017-05-16 12:49:55.000000000 +0000  
Change: 2017-05-18 00:15:09.161767445 +0000  
 Birth: -  
cssmuadm@lnx:~$
```

# Exercise

1. Find file with name `"initrd_ex1"` (tip: use `2> /dev/null` to avoid "permission denied messages")
2. Copy the file to the `"initrd"` directory in your home folder
3. Uncompress the file using the following command: `gzip -dc initrd_ex1 | cpio -ivd`
4. Find the **folder** with max size among uncompressed folders. Rename the folder to `max`
5. In the folder `"max"` find the file with type `"ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked.."` and move the file to your home directory
6. Rename the file with its size in bytes
7. Remove `"initrd"` directory

# File permissions: user and group

```
nikita@lnx: ~  
nikita@lnx:~$ ls -l  
total 3384  
-rw-rw-rwx 1 nikita nikita 20  
-rw-rw-r-- 1 nikita nikita 1  
-rw-rw-r-- 1 nikita nikita
```

user

group

# File permissions: ugo rwx

U G O

rwx rwx rwx

**Remember Mr. UGO**

```
nikita@lnx: ~  
nikita@lnx:~$ ls -l  
total 3384  
-rw-rw-rwx 1 nikita nikita 20  
-rw-rw-r-- 1 nikita nikita 1  
-rw-rw-r-- 1 nikita nikita
```

- U: user
- G: group
- O: other
- r: read
- w: write
- x: execute

# File permissions

**chmod** - change file mode bits:

Can use either numbers or symbols (u/g/o/a += rwx)

- **-R** - recursively

```
nikita@lnx: ~  
nikita@lnx:~$ ls -l  
total 2188  
----- 1 nikita nikita 2098176 Jul 22 2017 dll_libeay32  
-rw-rw-r-- 1 nikita nikita 119194 Nov 26 17:30 pdf_file  
-rw-rw-r-- 1 nikita nikita 10550 Sep 14 14:06 png_file  
drwxr-x--- 3 root root 4096 Jan 10 20:27 test  
nikita@lnx:~$ head dll_libeay32  
head: cannot open 'dll_libeay32' for reading: Permission denied  
nikita@lnx:~$ chmod u=rwx,g+rx,o=x dll_libeay32  
nikita@lnx:~$ ls -l  
total 2188  
-rwxr-x--x 1 nikita nikita 2098176 Jul 22 2017 dll_libeay32  
-rw-rw-r-- 1 nikita nikita 119194 Nov 26 17:30 pdf_file  
-rw-rw-r-- 1 nikita nikita 10550 Sep 14 14:06 png_file  
drwxr-x--- 3 root root 4096 Jan 10 20:27 test  
nikita@lnx:~$
```

# File permissions



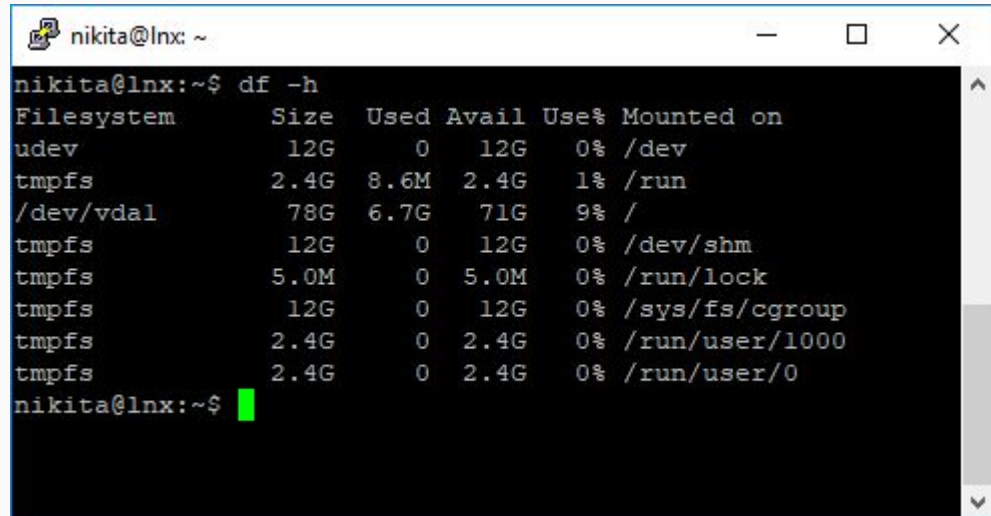
[\[link\]](#)

# No space on HDD?

**df** - report file system disk space usage

Useful flags:

- **-h**, --human-readable
- **-l**, --local limit listing to local file systems



```
nikita@lnx: ~  
nikita@lnx:~$ df -h  
Filesystem      Size  Used Avail Use% Mounted on  
udev            12G   0    12G   0% /dev  
tmpfs           2.4G  8.6M   2.4G   1% /run  
/dev/vda1       78G   6.7G   71G   9% /  
tmpfs           12G   0    12G   0% /dev/shm  
tmpfs           5.0M   0    5.0M   0% /run/lock  
tmpfs           12G   0    12G   0% /sys/fs/cgroup  
tmpfs           2.4G   0    2.4G   0% /run/user/1000  
tmpfs           2.4G   0    2.4G   0% /run/user/0  
nikita@lnx:~$
```



# How soft is the link?

**ln** - make links between files

- **-s** - create symbolic link rather than hard link (something like shortcut in Windows)

Use **unlink** command to remove a symlink

! Removing soft link does not remove the original file

! If all hard links removed, the file is removed

```
root@lnx: ~  
root@lnx:~# ln -s /bin/bash ./lback  
root@lnx:~# ls -l  
total 12  
drwxr-xr-x 2 root root 4096 May  2 14:45 dir1  
-rw-r--r-- 1 root root   0 May  2 14:34 file1  
lrwxrwxrwx 1 root root   9 May  2 15:57 lback -> /bin/bash  
-rwxrwxrwx 1 root root  78 Jun 23  2017 test_cpy.sh  
-rwxrwxrwx 1 root root  78 Jun 23  2017 test.sh  
root@lnx:~# rm lback  
root@lnx:~# ls -l /bin/ba*  
ls: cannot access '-': No such file or directory  
/bin/bash  
root@lnx:~# ls -l /bin/ba*  
-rwxr-xr-x 1 root root 1037528 May 16  2017 /bin/bash  
root@lnx:~# ln /bin/bash ./hback  
root@lnx:~# ls -l /bin/ba*  
-rwxr-xr-x 2 root root 1037528 May 16  2017 /bin/bash  
root@lnx:~# rm hback  
root@lnx:~# ls -l /bin/ba*  
-rwxr-xr-x 1 root root 1037528 May 16  2017 /bin/bash  
root@lnx:~#
```

Now there are 2 hard links!

# Exercise

1. Your home directory has bad permissions - it is writable/readable/executable by everybody. Change permissions to your home directory to the following:
  - a. You can do read/write/execute
  - b. Group members should be able to read and execute
  - c. Other users should be able to execute only
2. Create a new folder in your home directory with name “public”. Set the following permissions:
  - a. You: full permissions
  - b. Group members and other - read and execute only
3. Create few other folders in a new folder (e.g., 1, 2, 3). Can other users list contents of your home directory (using ls command)? Can other users see contents of your “/home/public” directory? Check with other students
4. In your “public” folder create a symbolic (soft) link to your home folder
5. Try to remove “execute” permissions for other users from your home directory. Can other students access your public folder now?