Introduction to the Linux OS

Peter Huszár

KFA: Department of Atmospheric Physics

Pavel Řezníček

ÚČJF: Institute of particle and nuclear physics

September 30, 2025

Quick list of Applications

Office Suites and Applications

Microsoft **Office** not available on Linux. However, there are number of alternatives:

- *LibreOffice* (former *OpenOffice*) fully included
- Gnome Office: *Abiword*, *Gnumeric*, *Gnucash*
- *Calligra Suite* (KDE focused office package)

PDF viewers:

- *Okular*: KDE PDF viewer, capable of adding annotations
- *Evince*: Gnome PDF viewer
- Old *Adobe reader* (needs special package repository)
- *Foxit reader*: not in package repositories, needs to be downloaded and installed manually

Latex packages naturally included in all distributions, with number of graphical editors allowing to debug Latex compilation and preview generated PDFs.

- *Texstudio*
- *Texmaker*
- *Kile* (KDE)
- *Visual Studio Code* + Latex plugin: free Microsoft product

Web-Browsers

The most **common browsers** are included:

- *Firefox*
- *Chrome* / Chromium
- *Opera*

There are also DE specific ones:

- *Konqueror* (KDE)
- *Epiphany* (Gnome)

As well as **terminal-based browsers** (limited use only, no javascripts etc.):

- *Lynx* / *Links* / *Elinks*
- *curl* / *wget*
- Useful for scripts to extract information from web

File-Browsers

Number of **file browsers**, usually specific to DE:

- *Dolphin* (KDE)
- *Nautilus* (Gnome)
- *Thunar* (XFce4)
- *Krusader*: KDE-based browser with *Total Commander* like look and features
- *Midnight commander*: terminal based 2-panel file browser (*Norton/Total Commander* like); very useful when working on remote machines as it does not need to transfer graphics

Terminal Emulators

Number of **terminal emulators**, usually specific to DE:

- *xterm*: basic terminal emulator
- *rxvt-unicode*: basic terminal emulator with unicode support
- *konsole*: KDE-based terminal emulator, support for tabs, complex configuration including predefined profiles etc.
- *gnome-terminal*: Gnome-based terminal emulator, support for tabs, complex configuration including predefined profiles etc.
- *xfce4-terminal*: Xfce4-based terminal emulator, support for tabs, complex configuration

Editors

Editors **for programmers** with various level of IDE:

- *Kate*: KDE based
- *KDevelop*: KDE based
- *Geany*: Gnome based
- *Eclipse*: heavy, Java based
- *Gedit*: light
- *Visual Studio Code*: Microsoft product, but open-source and free use (both home and commercial)

Console editors (useful when working on remote machines):

- *nano*: simple
- *jed*: simple
- *vim*: very powerful, but difficult for beginners
- *emacs*: very powerful, but difficult for beginners

Multimedia

Figures:

- *Gwenview* / *Gthumb* as picture viewers
- *Gimp* for bitmap drawings editing
- *Inkscape* for vector drawings editing

Video:

- *VLC*
- *SMplayer*
- *Xine*

Number of **command-line programs** for figures/audio/video conversions and processing

- *convert* for figures manipulations
- *ffmpeg* for audio/video manimulation

Emulators

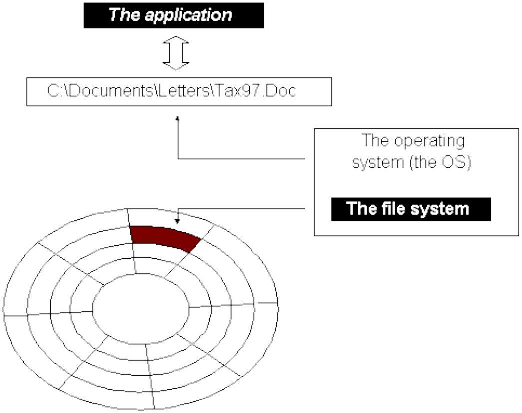
Emulators of **other (usually older) operating systems**:

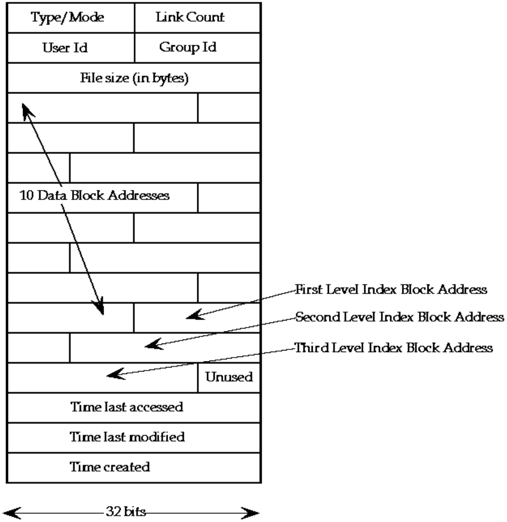
- *Wine*: emulator of Windows, many Windows applications able to run on that (not the complex ones as MS Office)
- *DosBox* / *DosEMU*: MS DOS emulators

Filesystems

Filesystem

System defining the way files and directories are stored on the physical disk





Command to show this file information: `stat`

Filesystems Overview

- FAT: *File Allocation Table*
  - The original MS-DOS filesystem, tree structure
  - Limited depth of sub-directories, limited filename lenght
  - Follow-up systems: FAT8/16/32, VFAT, exFAT
    - VFAT and exFAT are nowadays used on external/flash USB disks
- NTFS: *NT File System*
  - First appeared on Windows NT
  - Current MS filesystem on Windows
- ISO9600: Read-only files system for CD/DVD
  - Limited subdirectory length, limited filenames
  - Enhanced systems: Joliet, ElTorito, Rock Ridge

MS filesystems in Linux

- Linux can work with all of them (R/W).
- MS formats are not open $\Longrightarrow$ reverse engineering
- Support for the newest exFAT is yet somewhat experimental

Journaling filesystems on Linux

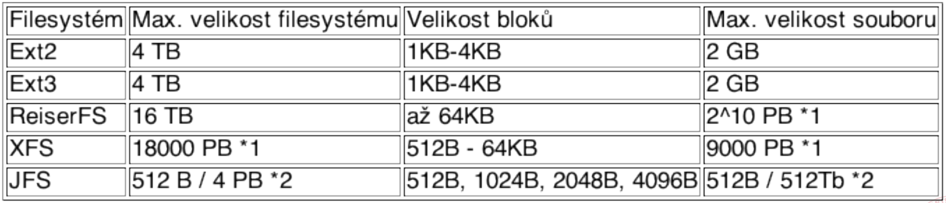
- Keeps track of disk operations not yet performed / finished
- Minimizing loss of data

Swap

- Enhancing RAM of the system
- When RAM gets full, swap is used to store temporarily unused data
- No special structure
- Windows equivalent: pagefile.sys

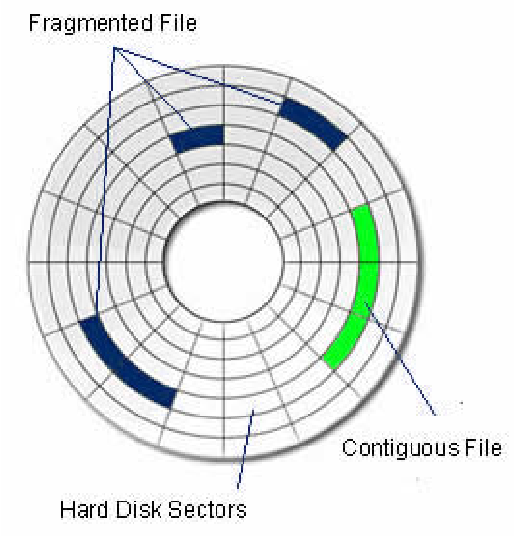
Filesystems in Linux

- `Ext2`, `Ext3`, `Ext4`: The most common on Linux
  - `Ext3` = `Ext2` + journal
  - `Ext4` allows for larger files and for more files per dir (32000 vs 64000)
- `XFS`: Silicon Graphics for the OS *IRIX*
  - Good performance for large data handling
- `ReiserFS`, `Reiser4`:
  - Good performance for large number of small files
  - Connecting advantages of filesystems and databases
  - Developed by Hans Raiser (in 2008 convicted murderer)a
- `JFS`: IBM for the OS *AIX* and later for *OS/2* and Linux
  - First journaling system
- `AFS`: *Andrew File System*
  - Distributed file system for large networks of servers and client workstations



Filesystém	Max. velikost filesystému	Velikost bloků	Max. velikost souboru
Ext2	4 TB	1KB-4KB	2 GB
Ext3	4 TB	1KB-4KB	2 GB
ReiserFS	16 TB	až 64KB	2^10 PB *1
XFS	18000 PB *1	512B - 64KB	9000 PB *1
JFS	512 B / 4 PB *2	512B, 1024B, 2048B, 4096B	512B / 512Tb *2

Fragmentation

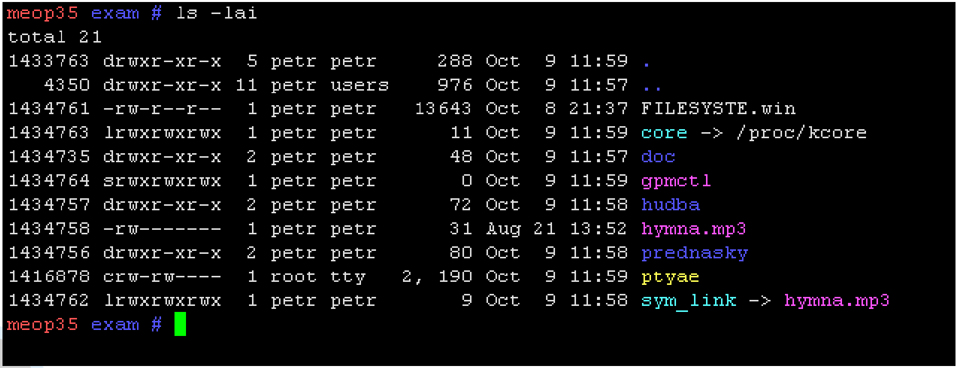


No defragmentation on Linux

- MS filesystems need occasional defragmentation:
  - Files are placed to the next free sector on the physical disk $\Longrightarrow$ when a file is enlarged, the next sector of physical disk is usually already occupied by another file $\Longrightarrow$ file fragmentation
- Linux filesystems defragment on the fly:
  - Files are placed on distant sections on the physical disk $\Longrightarrow$ space around existing file is usually free
  - If there is a danger file would become fragmented, the system tries to find a continuous space on the physical disk and moves the file in there
  - Fragmentation thus appears only when disk is close to full

Linux Files

- *Directory* (`d`): container of files
- *Plain file* (`-`): ordinary file (collection of bytes)
- *Symbolic link* (`l`): analogy of hypertext link or shortcut in Windows
- *Hard link*: similar to ordinary file, just another name for one file
- *Block and character device* (`b`,`c`): representation of HW devices
- *Name pipe* (`p`): named pipes for communication between applications instead through RAM
- *Socket* (`s`): duplex communication between processes



```
meop35 exam # ls -lai
total 21
1433763 drwxr-xr-x  5 petr petr     288 Oct  9 11:59 .
   4350 drwxr-xr-x 11 petr users    976 Oct  9 11:57 ..
1434761 -rw-r--r--  1 petr petr   13643 Oct  8 21:37 FILESYSTE.win
1434763 lrwxrwxrwx  1 petr petr      11 Oct  9 11:59 core -> /proc/kcore
1434735 drwxr-xr-x  2 petr petr      48 Oct  9 11:57 doc
1434764 srwxrwxrwx  1 petr petr       0 Oct  9 11:59 gpmctl
1434757 drwxr-xr-x  2 petr petr      72 Oct  9 11:58 hudba
1434758 -rw-------  1 petr petr      31 Aug 21 13:52 hymna.mp3
1434756 drwxr-xr-x  2 petr petr      80 Oct  9 11:58 prednasky
1416878 crw-rw----  1 root tty    2, 190 Oct  9 11:59 ptyae
1434762 lrwxrwxrwx  1 petr petr       9 Oct  9 11:58 sym_link -> hymna.mp3
meop35 exam #
```

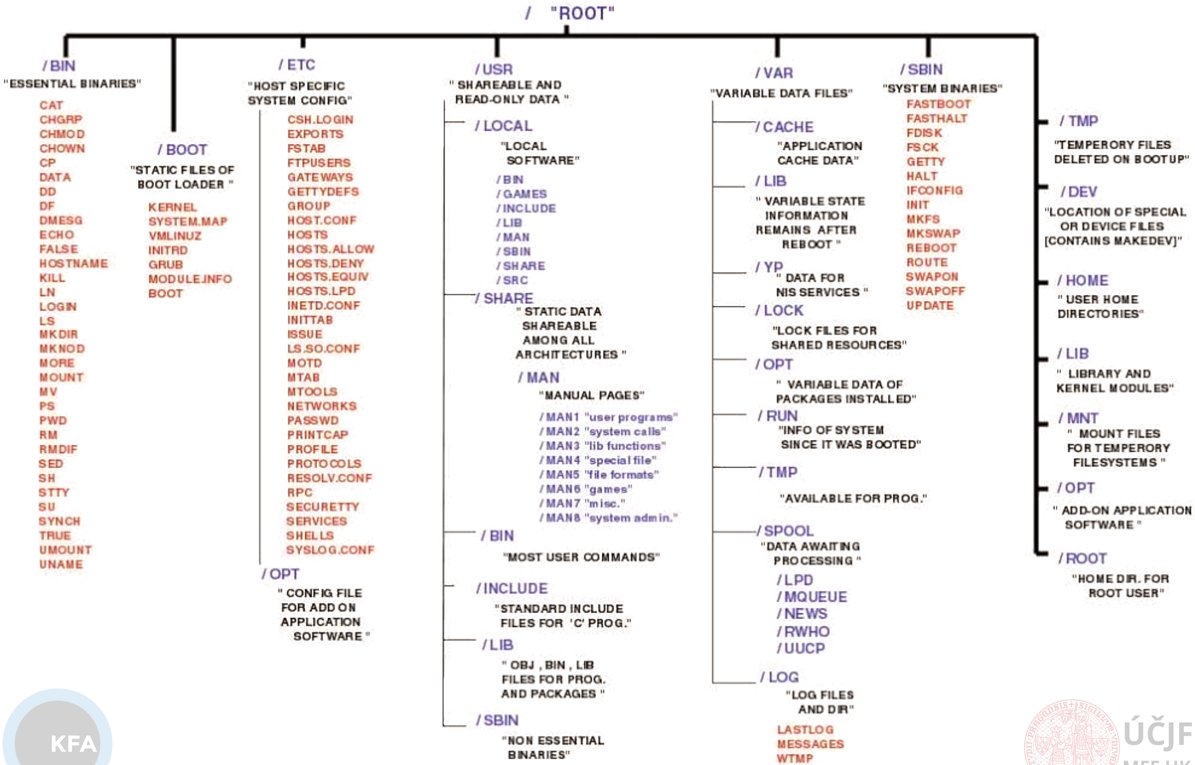
List files using command `ls -lah`

Structure of Directories in Linux

Directories and files in Linux

- Physical disks not visible in the structure (no `C:` and `D:`), disks are *mounted* to directories (admin can define what is mounted where)
  - e.g. typically "disk D:" is mounted to `/home` directory with users data
- Using '`/`' (slash) instead of Windows '`\`' (backslash)
- Linux filesystems (like `ext4`) features:
  - Directory and file names are CaSe SeNsItIvE
  - Native support of access control (user/group/all, read/write/execute)
  - Links (a bit similar to shortcuts in Windows, but for applications behave as real files)
- Hidden dirs and files starting with dot (`.*`, e.g. `.config`), usually keeping configuration of applications
- Special files representing e.g. HW devices and their configurations
- Special directory names:
  - Root directory '`/`' (top directory of the Linux system)
  - Home directory of a user '`~/`' (representing typically `/home/username`)
  - Current directory '`./`'
  - Directory one level up '`../`'
- **Absolute paths** starting from root of the system `/`
- **Relative paths** from current directory (may or may not start with '`./`')
- Text files has different standard for end-of-line (EOF) code from Windows/DOS/Old MACs
  - Conversion via commands `dos2unix` and `unix2dos`

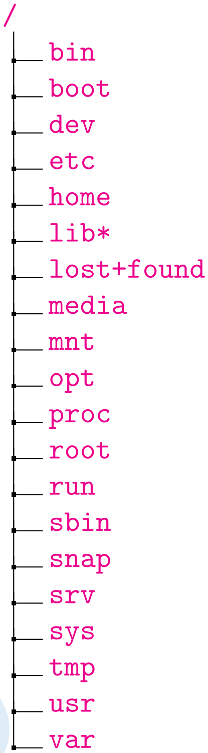
Diagram of the Directories



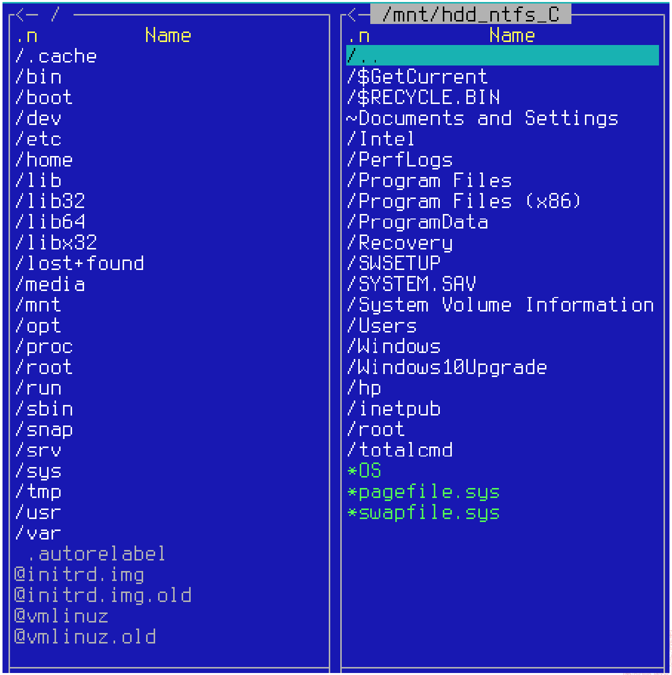
/ "ROOT"

- /BIN "ESSENTIAL BINARIES"
  - CAT, CHGRP, CHMOD, CHOWN, CP, DATA, DD, DF, DMESG, ECHO, FALSE, HOSTNAME, KILL, LN, LOGIN, LS, MKDIR, MKNOD, MORE, MOUNT, MV, PS, PWD, RM, RMDIF, SED, SH, STTY, SU, SYNCH, TRUE, UMOUNT, UNAME
- /BOOT "STATIC FILES OF BOOT LOADER "
  - KERNEL, SYSTEM.MAP, VMLINUZ, INITRD, GRUB, MODULE.INFO, BOOT
- /ETC "HOST SPECIFIC SYSTEM CONFIG"
  - CSH.LOGIN, EXPORTS, FSTAB, FTPUSERS, GATEWAYS, GETTYDEFS, GROUP, HOST.CONF, HOSTS, HOSTS.ALLOW, HOSTS.DENY, HOSTS.EQUIV, HOSTS.LPD, INETD.CONF, INITTAB, ISSUE, LS.SO.CONF, MOTD, MTAB, MTOOLS, NETWORKS, PASSWD, PRINTCAP, PROFILE, PROTOCOLS, RESOLV.CONF, RPC, SECURETTY, SERVICES, SHELLS, SYSLOG.CONF
  - /OPT " CONFIG FILE FOR ADD ON APPLICATION SOFTWARE "
- /USR " SHAREABLE AND READ-ONLY DATA "
  - /LOCAL "LOCAL SOFTWARE"
    - /BIN, /GAMES, /INCLUDE, /LIB, /MAN, /SBIN, /SHARE, /SRC
  - /SHARE " STATIC DATA SHAREABLE AMONG ALL ARCHITECTURES "
    - /MAN "MANUAL PAGES"
      - /MAN1 "user programs"
      - /MAN2 "system calls"
      - /MAN3 "lib functions"
      - /MAN4 "special file"
      - /MAN5 "file formats"
      - /MAN6 "games"
      - /MAN7 "misc."
      - /MAN8 "system admin."
  - /BIN "MOST USER COMMANDS"
  - /INCLUDE "STANDARD INCLUDE FILES FOR 'C' PROG."
  - /LIB " OBJ , BIN , LIB FILES FOR PROG. AND PACKAGES "
  - /SBIN "NON ESSENTIAL BINARIES"
- /VAR "VARIABLE DATA FILES"
  - /CACHE "APPLICATION CACHE DATA"
  - /LIB " VARIABLE STATE INFORMATION REMAINS AFTER REBOOT "
  - /YP " DATA FOR NIS SERVICES "
  - /LOCK "LOCK FILES FOR SHARED RESOURCES"
  - /OPT " VARIABLE DATA OF PACKAGES INSTALLED"
  - /RUN "INFO OF SYSTEM SINCE IT WAS BOOTED"
  - /TMP "AVAILABLE FOR PROG."
  - /SPOOL "DATA AWAITING PROCESSING "
    - /LPD, /MQUEUE, /NEWS, /RWHO, /UUCP
  - /LOG "LOG FILES AND DIR"
    - LASTLOG, MESSAGES, WTMP
- /SBIN "SYSTEM BINARIES"
  - FASTBOOT, FASTHALT, FDISK, FSCK, GETTY, HALT, IFCONFIG, INIT, MKFS, MKSWAP, REBOOT, ROUTE, SWAPON, SWAPOFF, UPDATE
- /TMP "TEMPERORY FILES DELETED ON BOOTUP"
- /DEV "LOCATION OF SPECIAL OR DEVICE FILES [CONTAINS MAKEDEV]"
- /HOME " USER HOME DIRECTORIES"
- /LIB " LIBRARY AND KERNEL MODULES"
- /MNT " MOUNT FILES FOR TEMPERORY FILESYSTEMS "
- /OPT " ADD-ON APPLICATION SOFTWARE "
- /ROOT "HOME DIR. FOR ROOT USER"

Linux Root Directory

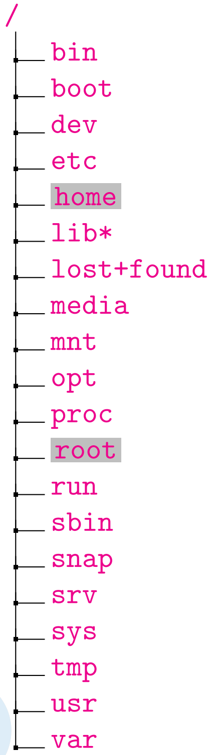


Snapshot of top directories in Linux and Windows 10:



```
<─ /                              <─ /mnt/hdd_ntfs_C
.n          Name                  .n          Name
/.cache                           /..
/bin                              /$GetCurrent
/boot                             /$RECYCLE.BIN
/dev                              ~Documents and Settings
/etc                              /Intel
/home                             /PerfLogs
/lib                              /Program Files
/lib32                            /Program Files (x86)
/lib64                            /ProgramData
/libx32                           /Recovery
/lost+found                       /SWSETUP
/media                            /SYSTEM.SAV
/mnt                              /System Volume Information
/opt                              /Users
/proc                             /Windows
/root                             /Windows10Upgrade
/run                              /hp
/sbin                             /inetpub
/snap                             /root
/srv                              /totalcmd
/sys                              *OS
/tmp                              *pagefile.sys
/usr                              *swapfile.sys
/var
 .autorelabel
@initrd.img
@initrd.img.old
@vmlinuz
@vmlinuz.old
```

User Home Directory



/
- bin
- boot
- dev
- etc
- home
- lib*
- lost+found
- media
- mnt
- opt
- proc
- root
- run
- sbin
- snap
- srv
- sys
- tmp
- usr
- var

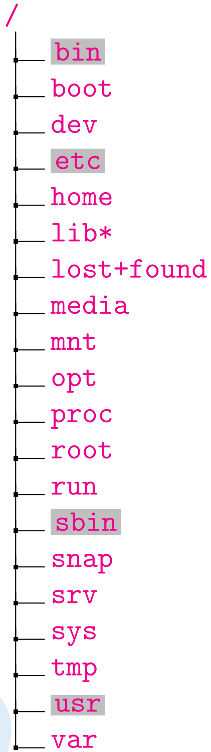
/home

- Home directory of users in `/home/$(USER)`
- Place used by all ordinary users to store data, configurations, possibly also local applications
- Other users can usually read files of other users, unless manually disabled
- `/home/$(USER)/Desktop`: Any file/directory placed here will appear on GUI desktop
- `/home/$(USER)/.*`: User configuration files of various applications
  - Especially in `.config` subdirectory
  - Files associations, user shortcuts etc. in `.local/share/` (`applications`) subdirectory
- Analogy of `C:\Users`, resp. `C:\Documents and Settings` in Windows

/root

- Home directory of administrator (*superuser*/*root* user)
- Same structure as for users in `home/$(USER)` directory

Application Installation



/
- bin
- boot
- dev
- etc
- home
- lib*
- lost+found
- media
- mnt
- opt
- proc
- root
- run
- sbin
- snap
- srv
- sys
- tmp
- usr
- var

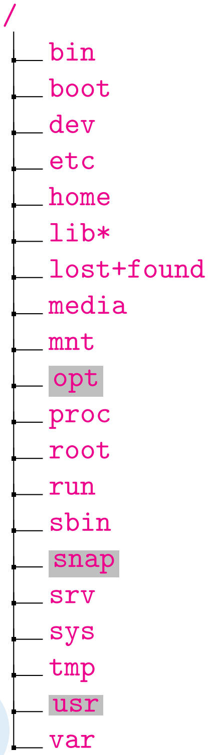
Ordinary application files

- Executable files in /usr/bin
- Data files in /usr/share
- Documentation, licence, examples etc. in /usr/share/doc
- Manual pages (help) for executables in /usr/share/man
- Libraries in /usr/lib*
- Header files of libraries in /usr/include

System application files

- Executable files in /bin
- *Superuser* executable files in /sbin and/or /usr/sbin
- /etc: Configuration of the system (services) and applications, common for all users
  - etc/default: Most common default settings
  - Analogy of Windows registry

Application Installation (Special, Local)


- /
  - bin
  - boot
  - dev
  - etc
  - home
  - lib*
  - lost+found
  - media
  - mnt
  - opt
  - proc
  - root
  - run
  - sbin
  - snap
  - srv
  - sys
  - tmp
  - usr
  - var

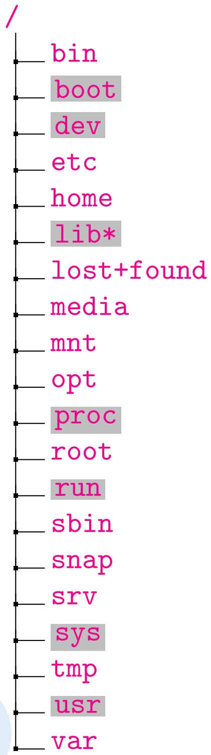
Local installations

- Usually not going through the ordinary package system, but directly copied to the system
- These programs and libraries placed in `/usr/local`
  - Contains similar sub-structure as the root `/` and/or `/usr` directories
  - I.e.: `etc`, `bin`, `lib`, `share`, ...

3rd party / commercial applications

- Whole application stored in `/opt`, with links to `/usr/bin` etc.
- `/snap` stores distribution-agnostic applications
  - All dependencies (libraries etc.) are included

Boot, Linux kernel, devices



/
- bin
- boot
- dev
- etc
- home
- lib*
- lost+found
- media
- mnt
- opt
- proc
- root
- run
- sbin
- snap
- srv
- sys
- tmp
- usr
- var

Boot configuration in /boot

- Linux kernels to boot
- Configuration of the boot manager (*GRUB*)

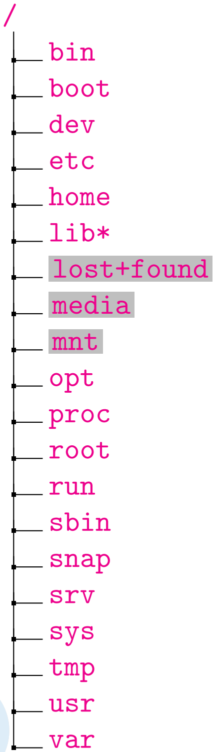
Linux kernel modules and source

- Linux kernels sit in /boot
- Linux kernel modules in /lib/modules
- Linux kernel source code in /usr/src

Hardware devices and processes

- Connected HW devices represented by special files in /dev
  - Storage devices to mount
  - Dustbin /dev/null
  - Console, random generator device, ...
- Configuration of the HW devices in /sys, *superuser* can use the files to modify the configuration of the HW drivers
- /proc directory keeps information about running processes, HW configuration (memory, cpu)
  - Interface to the kernel internal data structures
- Runtime information about processes in /run (e.g. connected WiFi)

Storage devices



/
- bin
- boot
- dev
- etc
- home
- lib*
- lost+found
- media
- mnt
- opt
- proc
- root
- run
- sbin
- snap
- srv
- sys
- tmp
- usr
- var

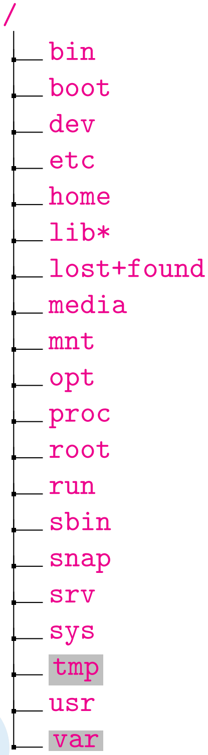
Lost files

- Lost and recovered files after OS crash

Storage devices connected to the system

- Removable devices usually appear in `/media`, resp. `/media/$(USER)`
  - Modern distributions connected them (*mount*) automatically
- Windows partitions, network storages etc. are usually mounted to `/mnt`
  - Follow rules defined in `/etc/fstab` file
  - And/or rules for auto-mounting in `/etc/auto*` files

Temporary files



(Semi-temporary application settings in /var

- /var/cache: Temporary files of applications, e.g.:
  - Downloaded installation packages of apt system in /var/cache/apt/archives
- /var/lib: Variable system information, e.g.:
  - Network connections
  - Installed packages info in /var/lib/dpkg/info
  - Content of package repositories in /var/lib/apt/lists
- /var/spool: Data awaiting further processing, e.g.:
  - E-mails
  - Print jobs
  - Scheduled user tasks
- /var/log: System log files:
  - Crucial for administrators
  - Boot, kernel messages, logins, services (e.g. WWW servers) etc.
- /var/www: WWW server data
- /var/lock: Application and system lockers

Temporary files in /tmp

- Temporary files of applications, e.g.:
  - Opened / downloaded files from web-browsers
  - Archives unzipped in file-browsers

Work with the Command Line in Terminals

Command Line

- Very powerfull in Linux, allows to master the OS
- Commands are taken by a *shell* and given to the OS
- Allows complicated scripts including loops, macros, conditions, etc.
- Several *shells* exists:
  - `bash`: most commont
  - `dash`: minimalistic, for system scripts
  - `zsh`: programmers focused
  - `ksh`:
  - `csh`: different syntax from bash-like shells above
  - `tcsh`: enhances csh
  - `fish`: friendly interactive shell
- `chsh` command to change user shell

Shortcut Keys

- **Left/Right/Home/End**: navigate cursor through the command line
- **Up/Down-Arrows**: browsing through history of commands
- **Tabulator**: complements commands or file names
  - Search for commands in the standard executable paths
  - Shows all possibilities in case the completion is ambiguous
  - Possibility to enhance completion for specific commands (ssh and remote host names etc.)
- **Ctrl-r**: search in history of commands backward
- **Ctrl-s**: search in history of commands forward
- **Ctrl-g**: end of search (not-only search) mode
- **Ctrl-l**: clear terminal window
- **Ctrl-q**: unblocks blocked terminal
  - Some terminal emulators get blocked with the Ctrl-s command
- **Ctrl+c**: interrupt running process
- **Ctrl+d**: interrupt writing into file (e.g. in `cat > filename`)

Shell Settings

- Configuration / startup files:
  - /.bash_history: history of commands
  - /.bashrc: startup script for non-login interactive shells
  - /.bash_profile: startup script for login shells (login in text console or from remote host)
    - /.profile is read by bash too, to be backward-compatible with old sh
  - /etc/profile: system-wide
  - /etc/motd: message on login shells
- Other shells have similarly-named files
- Typical configurations in /.bashrc:
  - Format of command prompt: PS1 (also PS2-PS4 for 2nd etc. level of prompts)
  - PATH to executables
  - PATH to 3rd party or local libraries (LD_LIBRARY_PATH)
  - Enhanced completion and other plugins (e.g. list of commands in not-installed packages)
  - Aliases (alias/unalias commands)
  - Default limits (ulimit command)
  - Environmental variables / program setups (export | less)
  - History length