Structure of Directories in Linux

KFA
MFF UK

ÚČJF
MFF UK

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 '`./`')
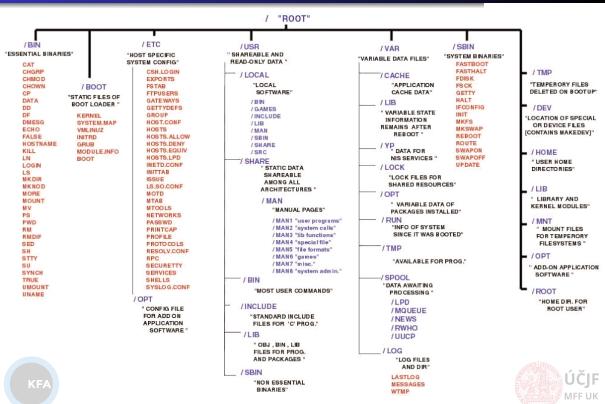- Text files has different standard for end-of-line (EOF) code from Windows/DOS/Old MACs
  - Conversion via commands `dos2unix` and `unix2dos`

KFA
MFF UK

ÚČJF
MFF UK

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

/
— bin
— boot
— dev
— etc
— home
— lib*
— lost+found
— media
— mnt
— opt
— proc
— root
— run
— sbin
— snap
— srv
— sys
— tmp
— usr
— var
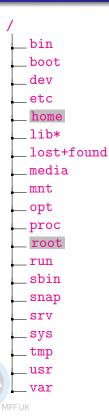
Snapshot of top directories in Linux and Windows 10:

```
<- /                        <- /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
```

KFA

MFF UK

ÚČJF
MFF UK

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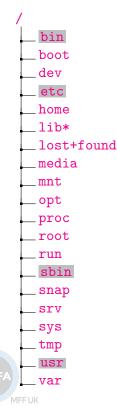
/home

- 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`
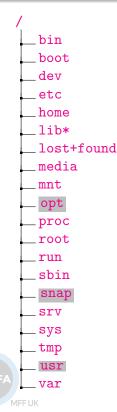- Libraries in `/usr/lib*`
- Header files of libraries in `/usr/include`

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`
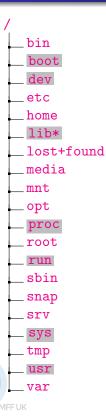 - Contains similar sub-structure as the root `/` and/or `/usr` directories
  - I.e.: `etc`, `bin`, `lib`, `share`, ...

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

ÚČJF
MFF UK

KFA
MFF UK

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
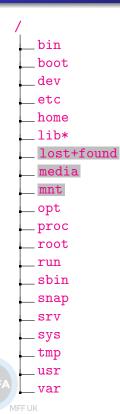- Linux kernel modules in /lib/modules
- Linux kernel source code in /usr/src

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

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

(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

KFA

MFF UK

ČJF
FF UK

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`)

KFA
MFF UK

ÚČJF
MFF UK

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

Commands in Linux

Programs available in shell or in the executable PATHs.

- General structure: `[path/ ]command [options ] [arguments ]`
- List of options and arguments (e.g. for `systemctl` command):
  - `command --help`, resp. `command -h`: basic syntax and options of the command
  - `man command`: manual pages with more detailed info
  - `info command`: interactive manual pages (jump to references etc.)
  - `command -v`. resp. `command --verbose`: be verbose about what is being done
- Run in background, while continuing work in shell: `command ... &`
- `fg`: Put command already running in background back to foreground of the shell
- **Ctrl+z**: Suspend process / command (stops processing)
- `bg`: Put command into background (typically used to let suspended command running again in the background)
- When command / program ends, it usually returns a *return code*
  - 0 ... usually means "no error"
  - non-zero is usually indication of premature exit of the command because of some problem
- Find full path to a command: `which command` (or search for it in aliases: command `alias` lists existing aliases)