

Unix Fundamentals – Files and Filesystems

Marek Kozłowski

Faculty of Mathematics and Information Sciences

Warsaw University of Technology

*Never use copy&paste for the following exercises. Retype all commands manually!
Don't just read those exercises and examples. Do them, please!*

1. *i-nodes*

- (a) Display *i-nodes* for files of different types, for example:

```
$ stat /etc/passwd
$ stat /etc/mtab
$ stat /etc
$ stat /dev/sda
$ stat /dev/tty
```
- (b) Display *i-numbers* for files and directories in your home:

```
$ ls -li
```
- (c) Create a non-empty file:

```
$ echo "something" > file1c1
```

Hard-link it with another name:

```
$ ln file1c1 file1c2
```

Check their *i-numbers*:

```
$ ls -li file1c?
```

Remove the first name and check if the file exists:

```
$ rm file1c1
$ cat file1c2
```
- (d) Create a non-empty file and display its *i-node* information:

```
$ echo "something" > file1d
$ stat file1d
```

Note that there are ‘change’ and ‘modification’ timestamps. What’s the difference?
Change group of (or permissions to) the file and display its *i-node* again:

```
$ chmod 644 file1d
$ stat file1d
```
- (e) Locate hard links under ‘/usr/bin’:

```
$ ls -li /usr/bin | awk '$3>1' | sort
```
- (f) List and try to identify (use ‘man 4’) character and block devices under ‘/dev’:

```
$ ls -l /dev
```

For example:

```
$ man 4 sd
$ man 4 tty
$ man 4 null
$ man 4 urandom
```
- (g) List block device files:

```
$ lsblk
$ ls -l /dev | grep -P '^b'
```
- (h) Cleanup:

```
$ rm file1*
```

2. Files

- (a) Create a directory and protect it from accidental deletion by 'rmdir':
\$ mkdir Dir2a
\$ touch Dir2a/.keep_me
- (b) What files from your home are currently open?
\$ lsof | grep \$HOME | grep -v cache
- (c) Create a file containing 128 random bytes:
\$ dd if=/dev/urandom of=file2c count=1 bs=128
\$ ls -l file2c
- (d) Display the total size of all elements in your home:
\$ du -sh ~
- (e) Cleanup:
\$ rm file2c; rm -r Dir2a

3. File systems

- (a) Take a look at Gentoo Linux notes on file systems:
https://wiki.gentoo.org/wiki/Handbook:AMD64/Installation/Disks#Creating_file_systems
- (b) Read about modern, feature rich file systems.
What are the advantages of *ext4*:
https://ext4.wiki.kernel.org/index.php/Main_Page
Why is *btrfs* so promising:
<https://btrfs.wiki.kernel.org/index.php/Main%5FPage>
Which systems use *HAMMER*:
<https://www.dragonflybsd.org/hammer/>
- (c) Find all executables (in '/usr/bin') hard-linked to 'mke2fs' – search for files with the same i-node numbers:
\$ ls -li /usr/bin/mke2fs
\$ ls -li /usr/bin | grep *paste_i-number_here*
Repeat the procedure for 'e2fsck'.
- (d) Check how many partitions are defined on your first HDD:
\$ ls -l /dev | grep sda
Plug some USB flash drive and check if your kernel recognizes it:
\$ ls -l /dev | grep sdb
- (e) List all filesystems currently mounted – check for their types and mounting options:
\$ mount
or
\$ cat /etc/mtab
On which partition does the root filesystem reside?
Does it use *ext3* or *ext4*?
- (f) Check free space for all mounted filesystems (for which the term *size* has any sense):
\$ df -h
- (g) Read man page on *mount*:
\$ man mount
Find what options can be set for all:
/FILESYSTEM-INDEPENDENT MOUNT OPTIONS
and peculiar file systems:
/FILESYSTEM-SPECIFIC MOUNT OPTIONS
- (h) What NFS shares and with which options are mounted as home directories at boot-up?
\$ cat /etc/fstab

- (i) Check if you can create a file system on a USB stick with a graphical application ('Disks' in desktop environment main menu). Can you create this file system manually (by calling 'mkfs')?
- (j) Read man page on software *Redundant Arrays of Independent Disks* (RAIDs) under Linux:
`$ man 4 md`
 What are the RAID types?
 Which RAID can be built of 2 drives? Do both of them offer fault tolerance?
 Which fault tolerant RAID models should be considered if there are three drives?
 How many drives can be corrupted in RAID6 without data loss?
- (k) Read about disk organization and naming conventions for other *NIX systems, for Solaris:
<https://docs.oracle.com/cd/E19455-01/805-6331/6j5vvgg680/index.html>
 and for FreeBSD:
https://docs.freebsd.org/en_US.IS08859-1/books/handbook/disk-organization.html
 Note that numerous Unices call fdisk partitions *slices*. The term *partition* is used for logical structures inside them (aka logical disks!).
- (l) Read (ask *Uncle Google* or *Auntie Wikipedia*) about data structure called a *B-tree*.
 What are other variants of this structure?
 Do you know any file systems that use those structures?

4. Filesystem hierarchy standard 2.x

- (a) Display the GRUB 1.x configuration. Does GRUB 1.x follow the KISS principle?
`$ less /boot/grub/menu.lst`
- (b) List kernel modules for your system's kernel:
`$ find /lib/modules/`uname -r`/kernel/ -name "*.ko.zst"`
- (c) Identify as many files in '/etc' discussed till now as possible. Take a more detailed look on shell-related files.
- (d) What is the home directory of *root*?
`$ grep root /etc/passwd`
- (e) Do you include 'stdio.h' in your source C files? Can you display it?
`$ less /usr/include/stdio.h`
- (f) Using file manager explore your icon themes folder and pixmaps folder. Those are:
`/usr/share/icons`
 and
`/usr/share/pixmaps`
 respectively. Take a look how icon *themes* are organized (it is a *freedesktop* standard).
- (g) Menu items for any desktop environment that follows *freedesktop* standards share the same syntax. Take a look at exemplary item description file:
`$ less /usr/share/applications/gvim.desktop`
- (h) What device files have their man pages:
`$ ls /usr/share/man/man4`
 or alternatively:
`$ find /usr/share/man -name "*.4.gz"`
- (i) Mount any USB flash drive using desktop icons. Under which directory is it mounted?
`$ mount | grep sdb`
- (j) How many processes are currently running?
`$ ls /proc | grep -E '^[0-9]*$' | wc -l`
- (k) What applications are installed in '/opt' on your computer?
- (l) Check for inconsistencies between *systemd* (your workstation) and the FHS described on last three slides.