MODERN OPERATING SYSTEMS

LECTURE 12

AUTHOR: DR. ZVEREVA OLGA M.

AGENDA

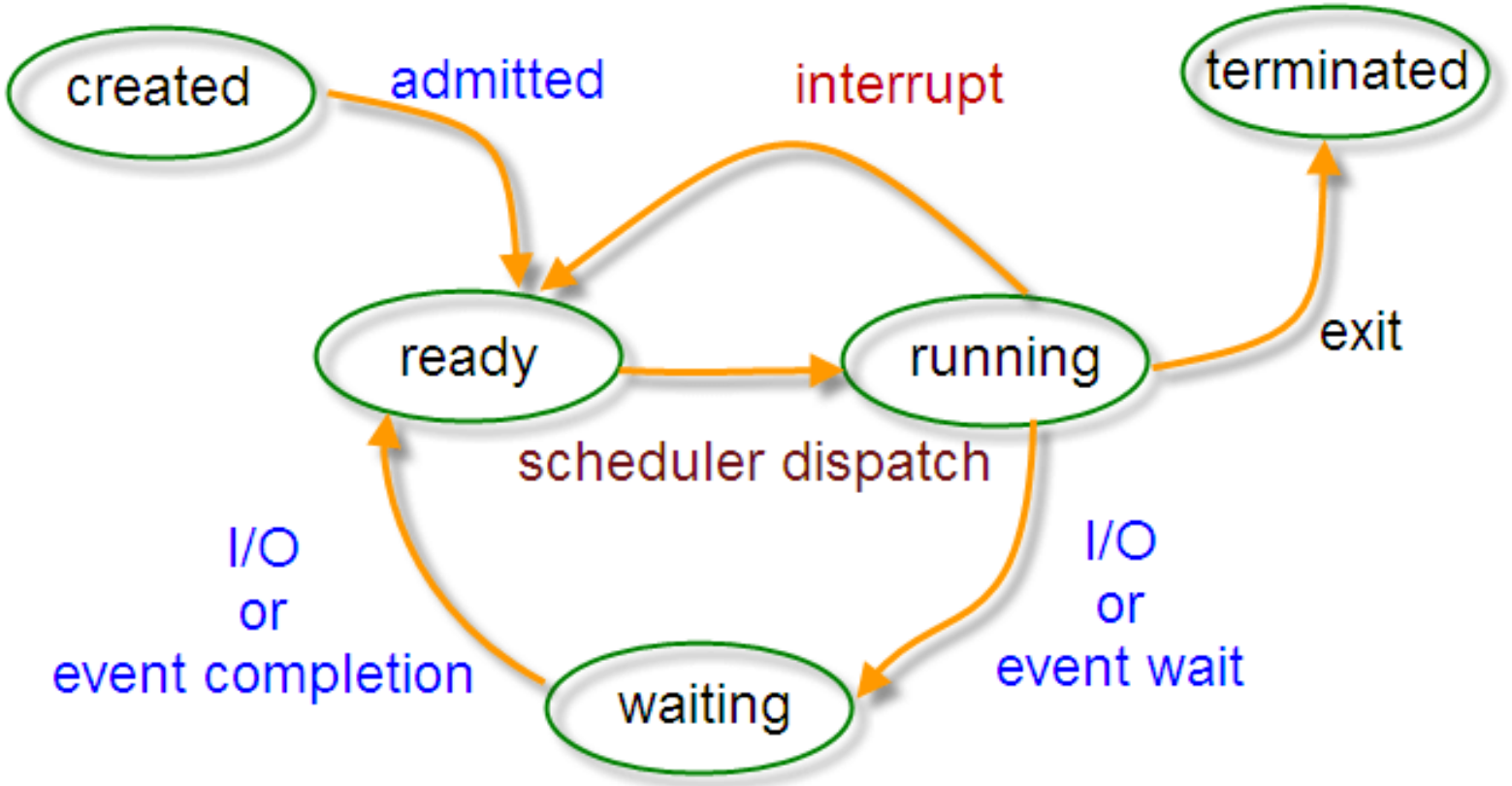
- PROCESSES IN LINUX
- FILE SYSTEMS IN LINUX
- FILE SYSTEM HIERARCHY STANDARD
- FILE TYPES
- COMMANDS FOR WORKING WITH FILES AND DIRECTORIES
- ACCESS PERMISSIONS
- CREATING AND MOUNTING FS

PROCESSES

PROCESS TYPES

- **FOREGROUND PROCESSES** (ALSO REFERRED TO AS INTERACTIVE PROCESSES) – THESE ARE INITIALIZED AND CONTROLLED THROUGH A TERMINAL SESSION. IN OTHER WORDS, THERE HAS TO BE A USER CONNECTED TO THE SYSTEM TO START SUCH PROCESSES; THEY HAVEN'T STARTED AUTOMATICALLY AS A PART OF THE SYSTEM FUNCTIONS/SERVICES.
- **BACKGROUND PROCESSES** (ALSO REFERRED TO AS NON-INTERACTIVE/AUTOMATIC PROCESSES) – ARE PROCESSES NOT CONNECTED TO A TERMINAL; THEY DON'T EXPECT ANY USER INPUT.
  - ✓ **DAEMONS** ARE SPECIAL TYPES OF BACKGROUND PROCESSES THAT START AT SYSTEM STARTUP AND KEEP RUNNING FOREVER AS A SERVICE; THEY DON'T DIE.

Process State



PROCESSES

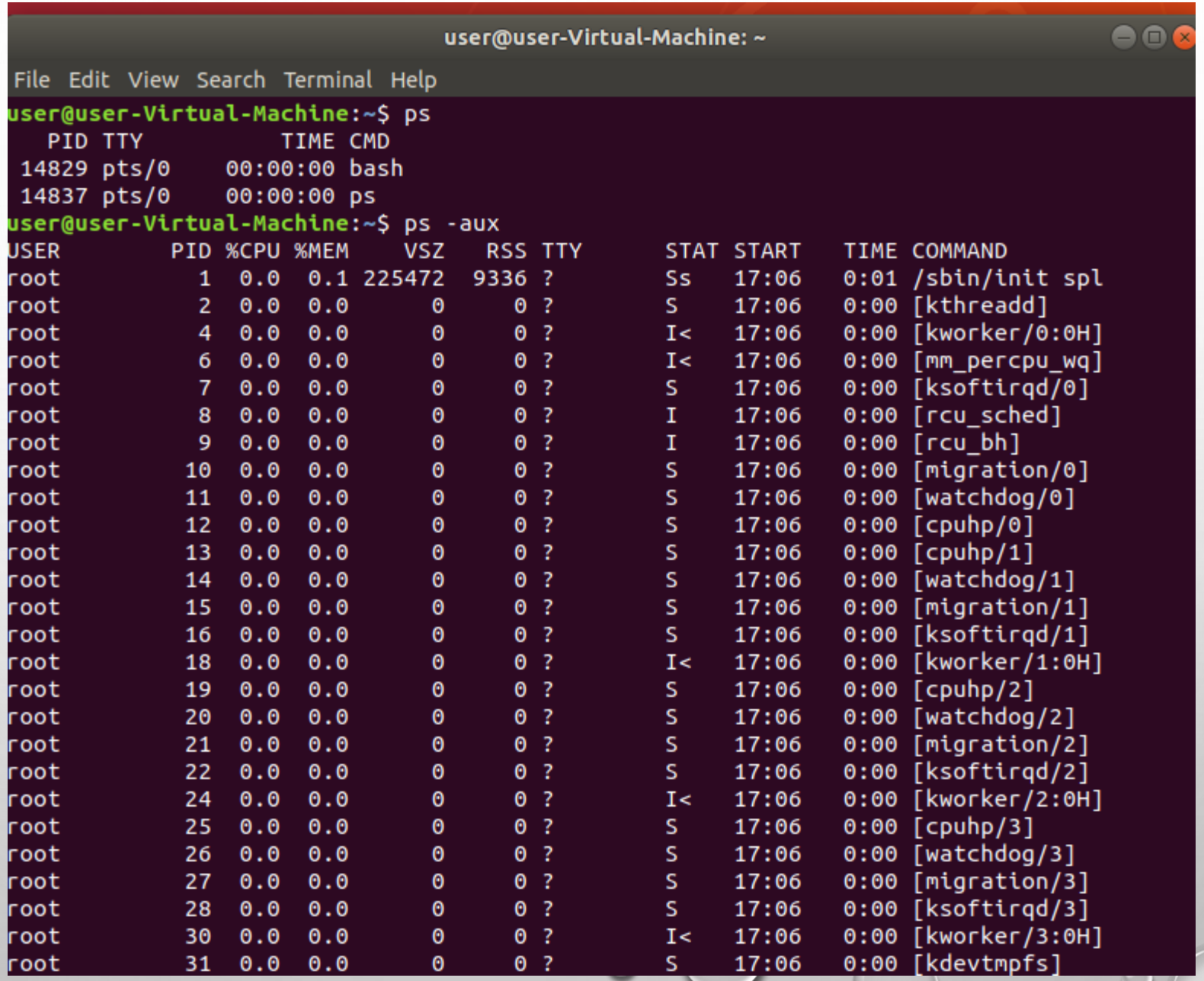
- A NEW PROCESS IS NORMALLY CREATED WHEN AN EXISTING PROCESS MAKES AN EXACT COPY OF ITSELF IN MEMORY. THE CHILD PROCESS WILL HAVE THE SAME ENVIRONMENT AS ITS PARENT, BUT ONLY THE PROCESS ID NUMBER IS DIFFERENT.
- A PROGRAM IS IDENTIFIED BY ITS PROCESS ID (**PID**) AS WELL AS IT'S PARENT PROCESSES ID (**PPID**), THEREFORE PROCESSES CAN FURTHER BE CATEGORIZED INTO:
  - ✓ **PARENT PROCESSES** – THESE ARE PROCESSES THAT CREATE OTHER PROCESSES DURING RUN-TIME.
  - ✓ **CHILD PROCESSES** – THESE PROCESSES ARE CREATED BY OTHER PROCESSES DURING RUN-TIME.
- **init** PROCESS (**systemd**) IS THE MOTHER (PARENT) OF ALL PROCESSES ON THE SYSTEM, IT'S THE FIRST PROGRAM THAT IS EXECUTED WHEN THE LINUX SYSTEM BOOTS UP; IT MANAGES ALL OTHER PROCESSES ON THE SYSTEM. IT IS STARTED BY THE KERNEL ITSELF, SO IN PRINCIPLE IT DOES NOT HAVE A PARENT PROCESS.

TO START A PROCESS

- ONCE YOU RUN A COMMAND OR PROGRAM (FOR EXAMPLE *process1*), IT WILL START A PROCESS IN THE SYSTEM. THIS WAY, YOU WILL START A FOREGROUND (INTERACTIVE) PROCESS, IT WILL BE CONNECTED TO THE TERMINAL AND A USER CAN SEND INPUT IT.
- TO START A PROCESS IN THE BACKGROUND USE SYMBOL "&" AT THE END OF ITS NAME (FOR EXAMPLE: process2&)
- TO SEND A BACKGROUND PROCESS TO THE FOREGROUND, USE THE **FG** COMMAND

HOW TO VIEW ACTIVE PROCESSES IN LINUX

- **ps** - COMMAND
- **top** – SYSTEM MONITORING TOOL
- **glances** – SYSTEM MONITORING TOOL
- **SYSTEM MONITOR** (IN ACTIVITIES – TOOL WITH GUI)


user@user-Virtual-Machine: ~
File Edit View Search Terminal Help
user@user-Virtual-Machine:~$ ps
PID TTY TIME CMD
14829 pts/0 00:00:00 bash
14837 pts/0 00:00:00 ps
user@user-Virtual-Machine:~$ ps -aux
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 1 0.0 0.1 225472 9336 ? Ss 17:06 0:01 /sbin/init spl
root 2 0.0 0.0 0 0 ? S 17:06 0:00 [kthreadd]
root 4 0.0 0.0 0 0 ? I< 17:06 0:00 [kworker/0:0H]
root 6 0.0 0.0 0 0 ? I< 17:06 0:00 [mm_percpu_wq]
root 7 0.0 0.0 0 0 ? S 17:06 0:00 [ksoftirqd/0]
root 8 0.0 0.0 0 0 ? I 17:06 0:00 [rcu_sched]
root 9 0.0 0.0 0 0 ? I 17:06 0:00 [rcu_bh]
root 10 0.0 0.0 0 0 ? S 17:06 0:00 [migration/0]
root 11 0.0 0.0 0 0 ? S 17:06 0:00 [watchdog/0]
root 12 0.0 0.0 0 0 ? S 17:06 0:00 [cpuhp/0]
root 13 0.0 0.0 0 0 ? S 17:06 0:00 [cpuhp/1]
root 14 0.0 0.0 0 0 ? S 17:06 0:00 [watchdog/1]
root 15 0.0 0.0 0 0 ? S 17:06 0:00 [migration/1]
root 16 0.0 0.0 0 0 ? S 17:06 0:00 [ksoftirqd/1]
root 18 0.0 0.0 0 0 ? I< 17:06 0:00 [kworker/1:0H]
root 19 0.0 0.0 0 0 ? S 17:06 0:00 [cpuhp/2]
root 20 0.0 0.0 0 0 ? S 17:06 0:00 [watchdog/2]
root 21 0.0 0.0 0 0 ? S 17:06 0:00 [migration/2]
root 22 0.0 0.0 0 0 ? S 17:06 0:00 [ksoftirqd/2]
root 24 0.0 0.0 0 0 ? I< 17:06 0:00 [kworker/2:0H]
root 25 0.0 0.0 0 0 ? S 17:06 0:00 [cpuhp/3]
root 26 0.0 0.0 0 0 ? S 17:06 0:00 [watchdog/3]
root 27 0.0 0.0 0 0 ? S 17:06 0:00 [migration/3]
root 28 0.0 0.0 0 0 ? S 17:06 0:00 [ksoftirqd/3]
root 30 0.0 0.0 0 0 ? I< 17:06 0:00 [kworker/3:0H]
root 31 0.0 0.0 0 0 ? S 17:06 0:00 [kdevtmpfs]

“TOP” TOOL

```
top - 19:55:24 up  2:48,  0 users,  load average: 0,23, 0,09, 0,04
Tasks: 243 total,   1 running, 198 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0,4 us,  0,0 sy,  0,0 ni, 99,6 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
KiB Mem :  6879248 total,  3714476 free,  1766104 used,  1398668 buff/cache
KiB Swap:  1048572 total,  1048572 free,        0 used.  4998144 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
 1402 user      20   0 4314396 561668  96712 S   0,7  8,2   4:49.11 gnome-shell
14959 user      20   0   45452   4224   3556 R   0,7  0,1   0:00.17 top
 1256 xrdp      20   0   57844  27060   6584 S   0,3  0,4   1:15.68 xrdp
 1272 user      20   0  709504  99916  44608 S   0,3  1,5   0:46.15 Xorg
    1 root      20   0  225472   9336   6776 S   0,0  0,1   0:01.64 systemd
    2 root      20   0       0      0      0 S   0,0  0,0   0:00.01 kthreadd
    4 root       0 -20       0      0      0 I   0,0  0,0   0:00.00 kworker/0:0H
    6 root       0 -20       0      0      0 I   0,0  0,0   0:00.00 mm_percpu_wq
    7 root      20   0       0      0      0 S   0,0  0,0   0:00.07 ksoftirqd/0
    8 root      20   0       0      0      0 I   0,0  0,0   0:00.46 rcu_sched
    9 root      20   0       0      0      0 I   0,0  0,0   0:00.00 rcu_bh
   10 root      rt   0       0      0      0 S   0,0  0,0   0:00.00 migration/0
   11 root      rt   0       0      0      0 S   0,0  0,0   0:00.02 watchdog/0
   12 root      20   0       0      0      0 S   0,0  0,0   0:00.00 cpuhp/0
   13 root      20   0       0      0      0 S   0,0  0,0   0:00.00 cpuhp/1
   14 root      rt   0       0      0      0 S   0,0  0,0   0:00.02 watchdog/1
   15 root      rt   0       0      0      0 S   0,0  0,0   0:00.00 migration/1
   16 root      20   0       0      0      0 S   0,0  0,0   0:00.03 ksoftirqd/1
   18 root       0 -20       0      0      0 I   0,0  0,0   0:00.00 kworker/1:0H
   19 root      20   0       0      0      0 S   0,0  0,0   0:00.00 cpuhp/2
   20 root      rt   0       0      0      0 S   0,0  0,0   0:00.02 watchdog/2
   21 root      rt   0       0      0      0 S   0,0  0,0   0:00.00 migration/2
   22 root      20   0       0      0      0 S   0,0  0,0   0:00.02 ksoftirqd/2
   24 root       0 -20       0      0      0 I   0,0  0,0   0:00.00 kworker/2:0H
   25 root      20   0       0      0      0 S   0,0  0,0   0:00.00 cpuhp/3
   26 root      rt   0       0      0      0 S   0,0  0,0   0:00.02 watchdog/3
   27 root      rt   0       0      0      0 S   0,0  0,0   0:00.00 migration/3
   28 root      20   0       0      0      0 S   0,0  0,0   0:00.02 ksoftirqd/3
   30 root       0 -20       0      0      0 I   0,0  0,0   0:00.00 kworker/3:0H
   31 root      20   0       0      0      0 S   0,0  0,0   0:00.01 kdevtmpfs
```

“GLANCES” TOOL

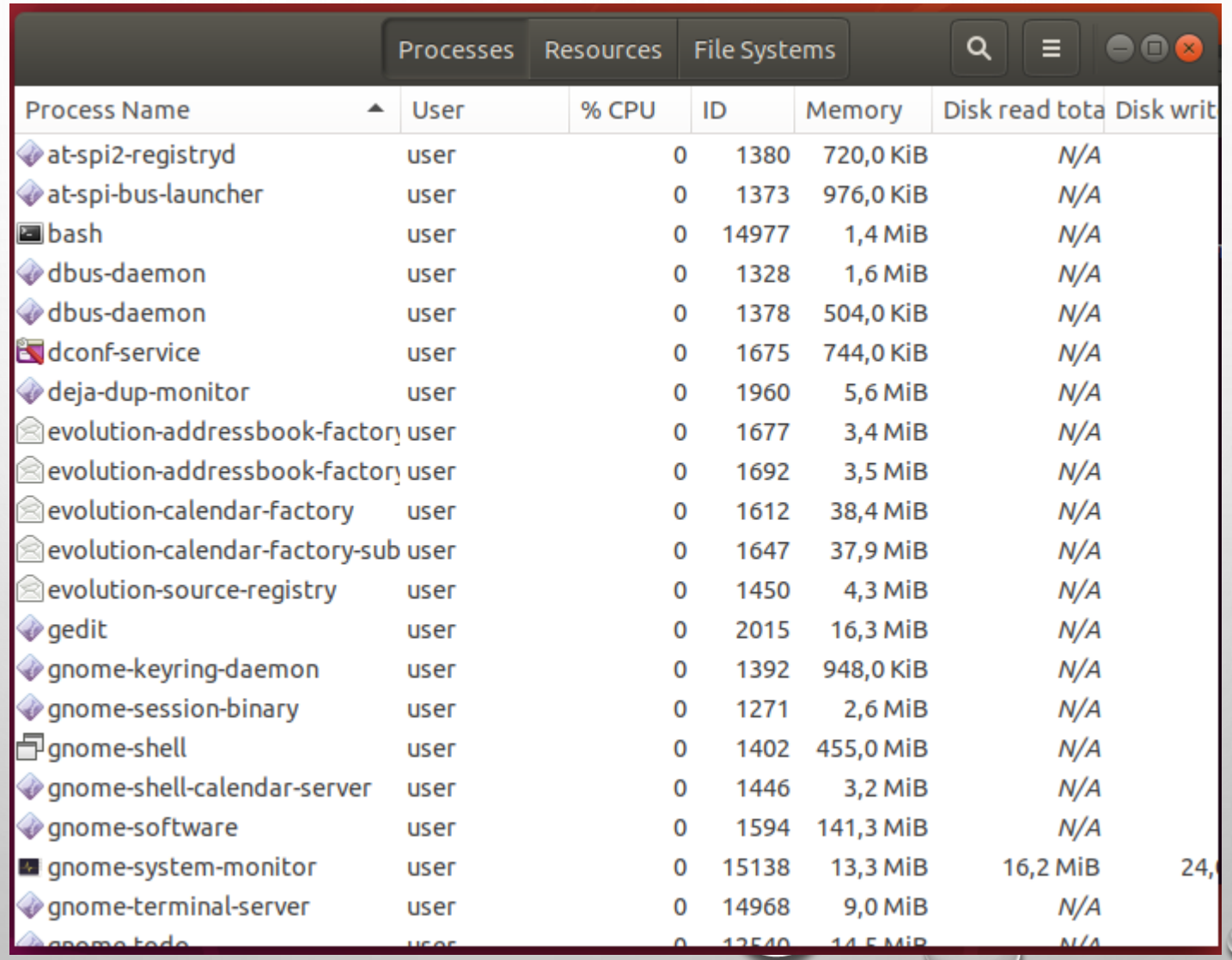
```
user-Virtual-Machine (Ubuntu 18.04 64bit / Linux 4.15.0-32-generic) - IP 172.20.184.37/31 Pub 212.193.78.43                    Uptime: 2:58:29

CPU  [|              8.4%]   CPU \     8.4%  nice:     6.2%  ctx_sw:   426   MEM -    25.6%  active:    1.87G   SWAP -     0.0%   LOAD    4-core
MEM  [|||            25.6%]   user:    1.1%  irq:      0.0%  inter:    148   total:   6.56G  inactive:   922M   total:   1024M   1 min:    0.12
SWAP [               0.0%]   system:  1.1%  iowait:   0.0%  sw_int:   434   used:    1.68G  buffers:    130M   used:       0   5 min:    0.17
                             idle:   91.6%  steal:    0.0%                  free:    4.88G  cached:    1.57G   free:    1024M   15 min:   0.09

NETWORK      Rx/s   Tx/s   TASKS 255 (600 thr), 1 run, 198 slp, 56 oth sorted automatically by cpu_percent, flat view
eth0           0b     0b
lo          256b   256b   Systemd          7    Services loaded: 227 active: 226 failed: 1

DefaultGateway      12ms      CPU%  MEM%  VIRT   RES    PID USER        NI S     TIME+   R/s   W/s Command
                               2.9   8.2 4.11G  550M   1402 user         0 S   5:57.30     0     0 /usr/bin/gnome-shell
DISK I/O       R/s    W/s      2.9   0.6  544M 38.5M  16986 user         0 R   0:01.62     0     0 /usr/bin/python3 /usr/bin/glances
sda1             0    43K      1.9   0.4 56.8M 26.7M   1256 xrdp         0 S   1:24.43     0     0 /usr/sbin/xrdp
sda14            0      0      1.0   1.5  693M 97.8M   1272 user         0 S   0:55.59     0     0 /usr/lib/xorg/Xorg :10 -auth .Xauthority -con
sda15            0      0      0.3   0.1  301M 8.24M    991 root         0 S   0:00.25     0     0 /usr/lib/upower/upowerd
                               0.3   0.4  508M 24.0M   1159 gdm          0 S   0:00.18     0     0 /usr/lib/gnome-settings-daemon/gsd-power
FILE SYS     Used   Total      0.3   0.1 1.24G 8.24M   1415 user         0 S   0:00.25     0     0 /usr/bin/pulseaudio --start --log-target=sysl
/ (sda1)   7.24G   10.5G      0.3   0.4  648M 23.9M   1541 user         0 S   0:00.82     0     0 /usr/lib/gnome-settings-daemon/gsd-color
/boot/efi  3.34M    104M      0.3   0.6  780M 37.4M  14968 user         0 S   0:02.58     0     0 /usr/lib/gnome-terminal/gnome-terminal-server
_ore/6531  91.1M   91.1M      0.0   0.1  221M 9.51M      1 root         0 S   1:53.38     0     0 /sbin/init splash
_ore/6673  89.2M   89.2M      0.0   0.0     0     0      2 root         0 S   0:00.10     0     0 kthreadd
_ore/6818  89.5M   89.5M      0.0   0.0     0     0      4 root       -20 ?   0:00.00     0     0 kworker/0:0H
_re18/941  53.8M   53.8M      0.0   0.0     0     0      6 root       -20 ?   0:00.00     0     0 mm_percpu_wq
_-1604/70   141M    141M      0.0   0.0     0     0      7 root         0 S   0:00.11     0     0 ksoftirqd/0
_-1604/78   141M    141M      0.0   0.0     0     0      8 root         0 ?   0:00.56     0     0 rcu_sched
_-1604/82   141M    141M      0.0   0.0     0     0      9 root         0 ?   0:00.00     0     0 rcu_bh
_-1804/36   151M    151M      0.0   0.0     0     0     10 root         0 S   0:00.00     0     0 migration/0
_-1804/40   151M    151M      0.0   0.0     0     0     11 root         0 S   0:00.20     0     0 watchdog/0
_ator/180  2.38M   2.38M      0.0   0.0     0     0     12 root         0 S   0:00.00     0     0 cpuhp/0
_ator/260  2.25M   2.25M      0.0   0.0     0     0     13 root         0 S   0:00.00     0     0 cpuhp/1
_ator/406  4.12M   4.12M      0.0   0.0     0     0     14 root         0 S   0:00.20     0     0 watchdog/1
_ters/139  13.0M   13.0M      0.0   0.0     0     0     15 root         0 S   0:00.00     0     0 migration/1
_ters/254  14.8M   14.8M      0.0   0.0     0     0     16 root         0 S   0:00.40     0     0 ksoftirqd/1
_ters/258  14.8M   14.8M      0.0   0.0     0     0     18 root       -20 ?   0:00.00     0     0 kworker/1:0H
_-logs/37  14.5M   14.5M
2019-05-14 20:05:26       No warning or critical alert detected
```

SYSTEM MONITOR



Process Name	User	% CPU	ID	Memory	Disk read tota	Disk writ
at-spi2-registryd	user	0	1380	720,0 KiB	N/A	
at-spi-bus-launcher	user	0	1373	976,0 KiB	N/A	
bash	user	0	14977	1,4 MiB	N/A	
dbus-daemon	user	0	1328	1,6 MiB	N/A	
dbus-daemon	user	0	1378	504,0 KiB	N/A	
dconf-service	user	0	1675	744,0 KiB	N/A	
deja-dup-monitor	user	0	1960	5,6 MiB	N/A	
evolution-addressbook-factory	user	0	1677	3,4 MiB	N/A	
evolution-addressbook-factory	user	0	1692	3,5 MiB	N/A	
evolution-calendar-factory	user	0	1612	38,4 MiB	N/A	
evolution-calendar-factory-sub	user	0	1647	37,9 MiB	N/A	
evolution-source-registry	user	0	1450	4,3 MiB	N/A	
gedit	user	0	2015	16,3 MiB	N/A	
gnome-keyring-daemon	user	0	1392	948,0 KiB	N/A	
gnome-session-binary	user	0	1271	2,6 MiB	N/A	
gnome-shell	user	0	1402	455,0 MiB	N/A	
gnome-shell-calendar-server	user	0	1446	3,2 MiB	N/A	
gnome-software	user	0	1594	141,3 MiB	N/A	
gnome-system-monitor	user	0	15138	13,3 MiB	16,2 MiB	24,
gnome-terminal-server	user	0	14968	9,0 MiB	N/A	

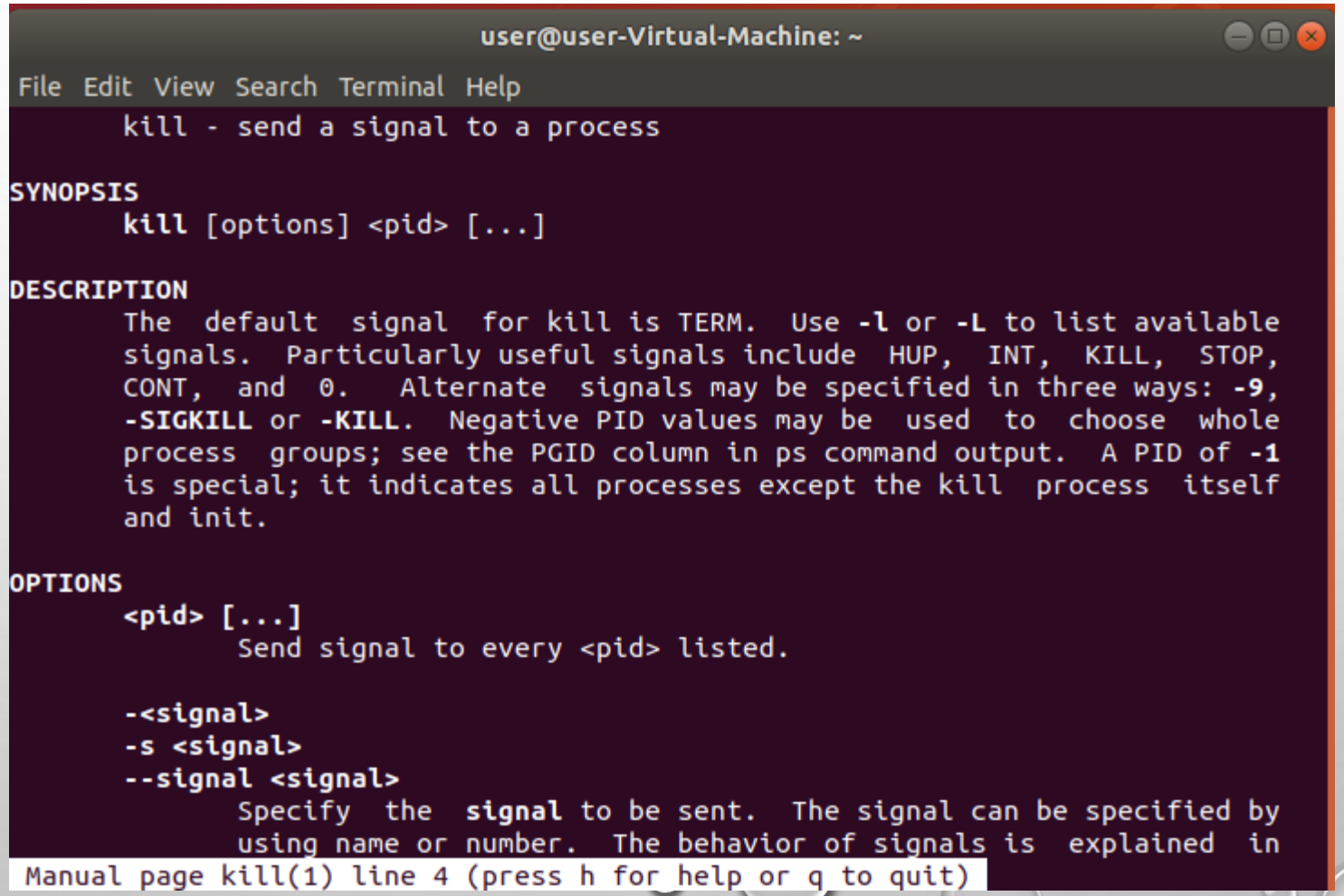
PROCESS PRIORITY

- ON THE LINUX SYSTEM, ALL ACTIVE PROCESSES HAVE A PRIORITY AND CERTAIN NICE VALUE.
- NICE VALUE CAN BE UNDERSTOOD AS A RELATIVE PRIORITY, YOU CAN CHANGE THE PROCESS PRIORITY BY CHANGING NICE VALUE (TO CHANGE NICE VALUE – COMMANDS “nice”, “renice”)
- PROCESSES WITH HIGHER PRIORITY WILL NORMALLY GET MORE CPU TIME THAN LOWER PRIORITY PROCESSES.

SIGNALS

- **SIGNAL** IS A NOTIFICATION, A MESSAGE SENT BY EITHER OPERATING SYSTEM OR SOME APPLICATION TO OUR PROGRAM. SIGNALS ARE A MECHANISM FOR ONE-WAY ASYNCHRONOUS NOTIFICATIONS. A SIGNAL MAY BE SENT FROM THE KERNEL TO A PROCESS, FROM A PROCESS TO ANOTHER PROCESS, OR FROM A PROCESS TO ITSELF. SIGNAL TYPICALLY ALERT A PROCESS TO SOME EVENT, SUCH AS A SEGMENTATION FAULT, OR THE USER PRESSING CTRL-C.
- LINUX KERNEL IMPLEMENTS ABOUT 30 SIGNALS. EACH SIGNAL IDENTIFIED BY A NUMBER, FROM 1 TO 31. SIGNALS DON'T CARRY ANY ARGUMENT AND THEIR NAMES ARE MOSTLY SELF EXPLANATORY. FOR INSTANCE **SIGKILL** OR SIGNAL NUMBER **9** TELLS THE PROGRAM THAT SOMEONE TRIES TO KILL IT.
- THE COMMAND **KILL** CAN BE USED TO SEND A SIGNAL TO A PROCESS OTHER THAN THE CURRENT FOREGROUND PROCESS.

COMMAND “kill”

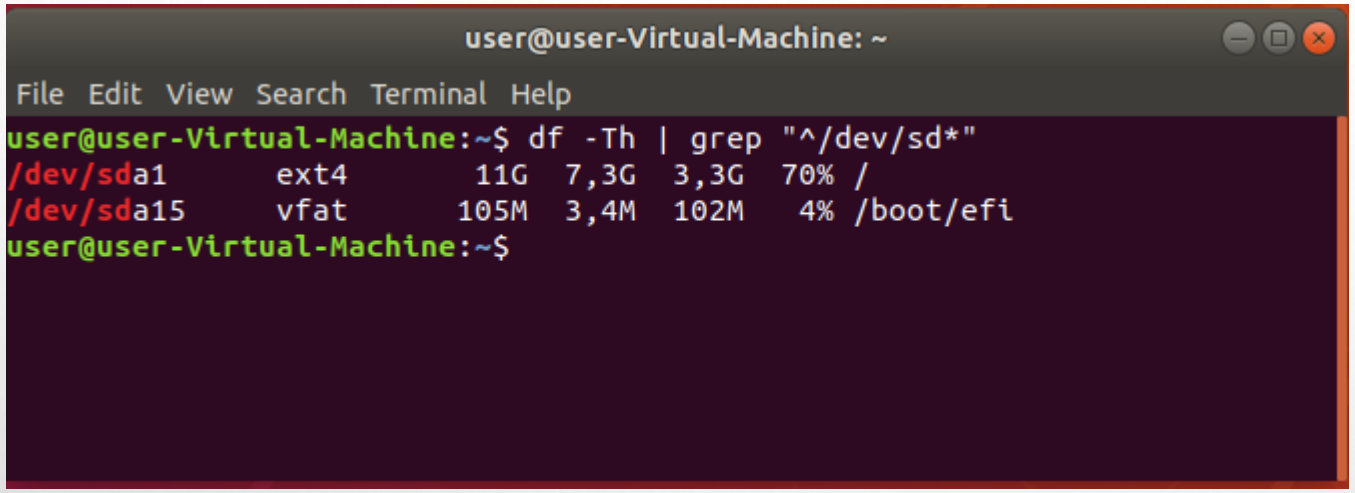


FILE SYSTEMS

FILE SYSTEMS UNDER LINUX

- THERE ARE SEVERAL TYPE FILE SYSTEMS DESIGNED TO WORK UNDER LINUX
  - ✓ EXT2FS – THE SECOND EXTENDED FILE SYSTEM (WORKED OUT BY REMY CARD IN 1993)
  - ✓ EXT3FS – THE THIRD EXTENDED FILE SYSTEM (AN EXTENSION OF THE EXT2FS, A JOURNAL TO STORE DATA ABOUT FILE SYSTEM MODIFYING OPERATIONS WAS INTRODUCED INTO IT, THE AUTHOR IS STEPHEN TWEEDIE, WAS ENGINEERED IN 1999). TODAY, THERE IS EXT4FS – A 64 BIT FS.
  - ✓ REISERFS (REISER3, REISER4), JFS, BTRFS (B-TREE FS *ИЛИ* BUTTER FS) , TUX2 & TUX3, AND ETC.
- TO DETERMINE WHICH FS IN INSTALLED IN EVERY PARTITION THERE SEVERAL WAY, ONE OF THEM TO USE “df –TH” COMMAND

“df” COMMAND



df – is a command which describes free space volumes, with “Th” options shows types of FS

“command1 | command2” – indicates command conveyer, which means that output of command1 becomes the input of the command2

“grep” – prints lines matching a pattern

Pattern “^/dev/sd*” is a regular expression. It means :

^ - at the beginning of line

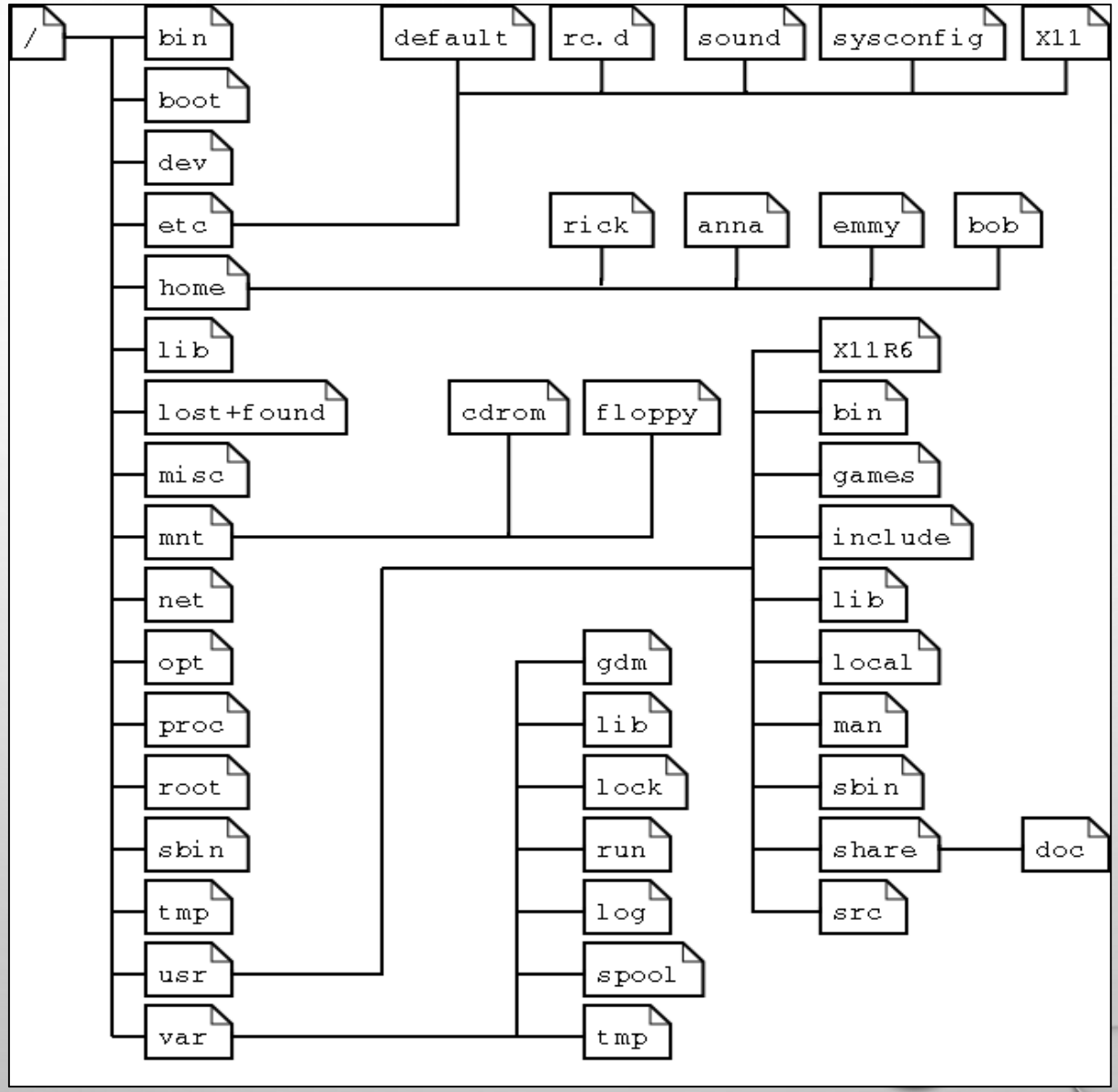
“*” – any sequence of symbols

In a whole: we are looking for a line in the output of “df” command which begins with the name of any disk (it relates to disk partition)

FILE SYSTEM HIERARCHY STANDARD

Directory name	Description of the content
/bin	This is where the executable files are located. These files are available to all users
/boot	Contains files for booting the system
/dev	This is where files associated with devices are located
/etc	Supervisor directory commands, configuration files, disk configuration files, valid user lists, groups, ethernet, hosts, where to send critical messages
/home	Contains the home directory for users and other accounts (except root)
/lib	Contains shared library files and sometimes other kernel-related files
/mnt	Used to mount other file systems (e.g. CD-ROM FS)
/proc	Contains all processes marked as a file by process number or other information that is dynamic to the system
/usr	Used for miscellaneous purposes. As s rule, various software environments are installed here
/var	Typically contains variable-length files such as log and print files and any other type of file that may contain a variable amount of data

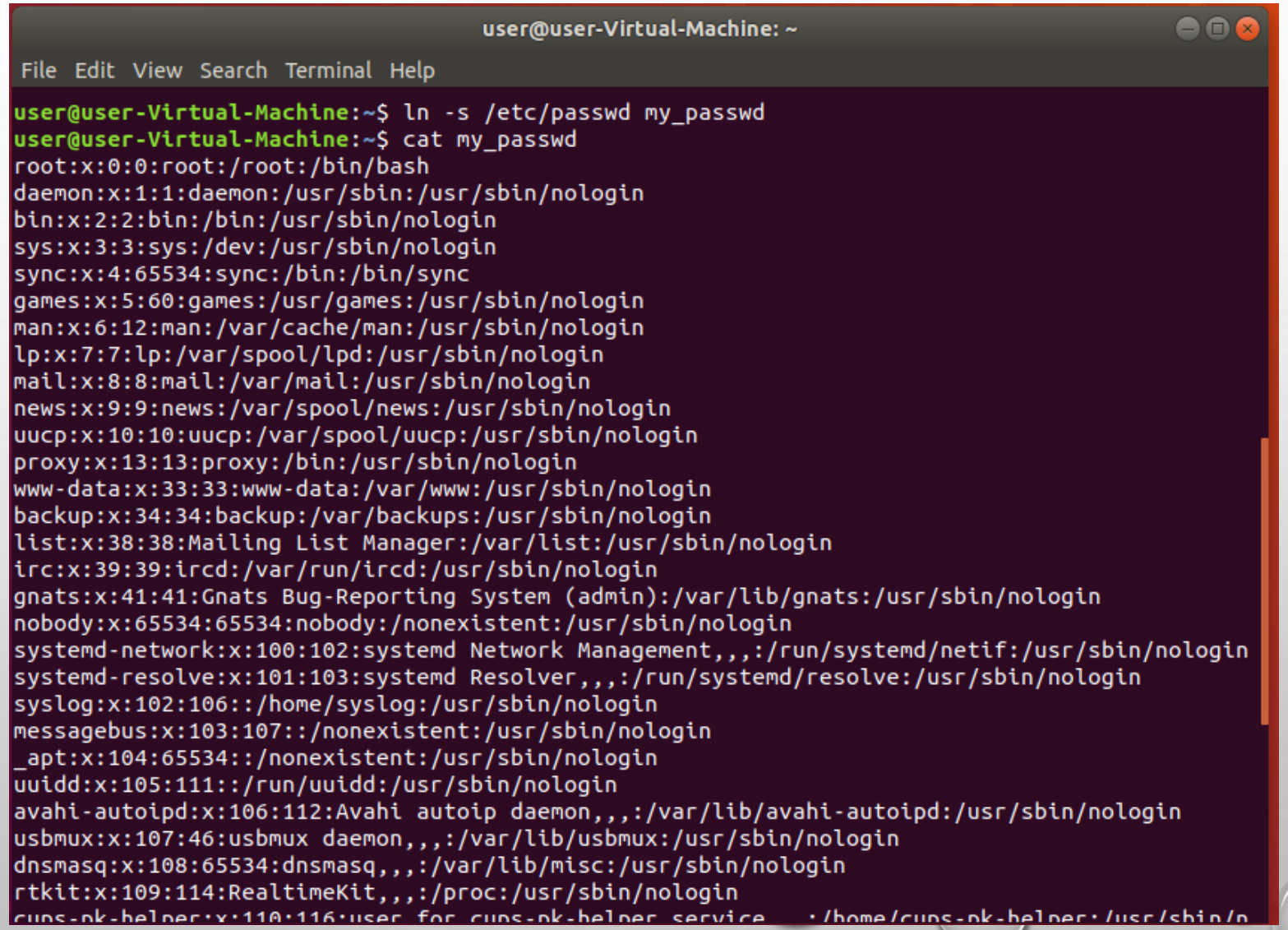
FILE SYSTEM HIERARCHY



LINKING FILES: SYMBOLIC VS HARD LINKS

- A LINK IS NOTHING MORE THAN A WAY OF MATCHING TWO OR MORE FILE NAMES TO THE SAME SET OF FILE DATA (NOT TO HAVE A LOT OF COPIES).
- THERE ARE TWO WAYS TO ACHIEVE THIS:
  - ✓ HARD LINK: ASSOCIATE TWO OR MORE FILE NAMES WITH THE SAME INODE. HARD LINKS SHARE THE SAME DATA BLOCKS ON THE HARD DISK, WHILE THEY CONTINUE TO BEHAVE AS INDEPENDENT FILES. IS MADE BY MEANS OF THE FILE SYSTEM CODE, NO NEW OBJECTS ARE CREATED IN THE FS.
  - ✓ HARD LINK SPECIFICITY:
    - ❖ A HARD LINK CAN'T SPAN PARTITIONS (TO REDIRECT TO ANOTHER FS), BECAUSE INODE NUMBERS ARE ONLY UNIQUE WITHIN A GIVEN PARTITION
    - ❖ IF A HARD LINK IS DELETED, DATA IS NOT DELETED FROM THE DISK IN CASE IT IS NOT THE LAST HARD LINK FOR THIS DATA
  - ✓ SOFT LINK (SYMLINK): A SMALL FILE THAT IS A POINTER TO ANOTHER FILE. A SYMBOLIC LINK CONTAINS THE PATH TO THE TARGET FILE INSTEAD OF A PHYSICAL LOCATION ON THE HARD DISK
  - ✓ SOFT LINK SPECIFICITY :
    - ✓ SINCE INODES ARE NOT USED IN THIS SYSTEM, SOFT LINKS CAN SPAN ACROSS PARTITIONS.
    - ✓ IF A SYMLINK IS DELETED, IT CAUSES NO EFFECT ON DATA STORED
    - ✓ IF ALL HARD LINKS TO DISK DATA ARE DELETED AND THERE IS A SYMLINK TO THIS DATA, IT BECOMES USELESS

CREATING A SYMLINK

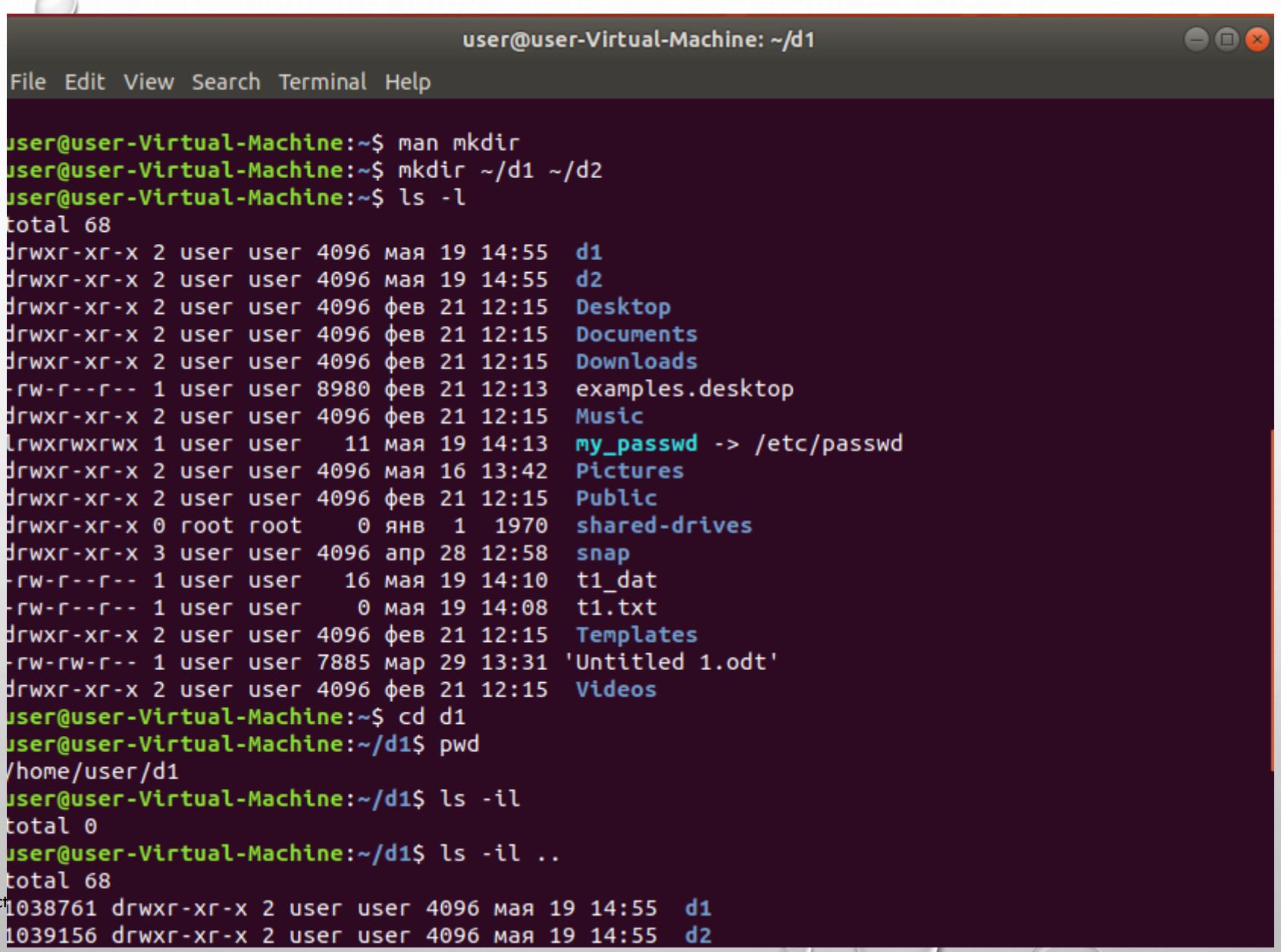


NECESSARY TO KNOW

- FILE NAMES ARE CASE SENSITIVE (“FILE1” AND “file1” ARE DIFFERENT NAMES)
- FILE NAMES DO NOT INCLUDE EXTENSIONS (NAME “FILE.TXT” DOES NOT INDICATE THAT THIS FILE STORES TEXT)
- EVERY FILE (DIRECTORY) IS DETERMINED BY ITS INODE
- THERE ARE TWO “SPECIFIC” FOR THE SYSTEM DIRECTORIES: HOME DIRECTORY (SPECIFIED AS “~”), AND THE CURRENT DIRECTORY (SPECIFIED AS “.”)
- IF A NAME INCLUDES BLANKS, IT IS NECESSARY SURROUND IT BY APOSTROPHES OR QUOTATION MARKS (“FILE NAME” OR ‘FILE NAME’)

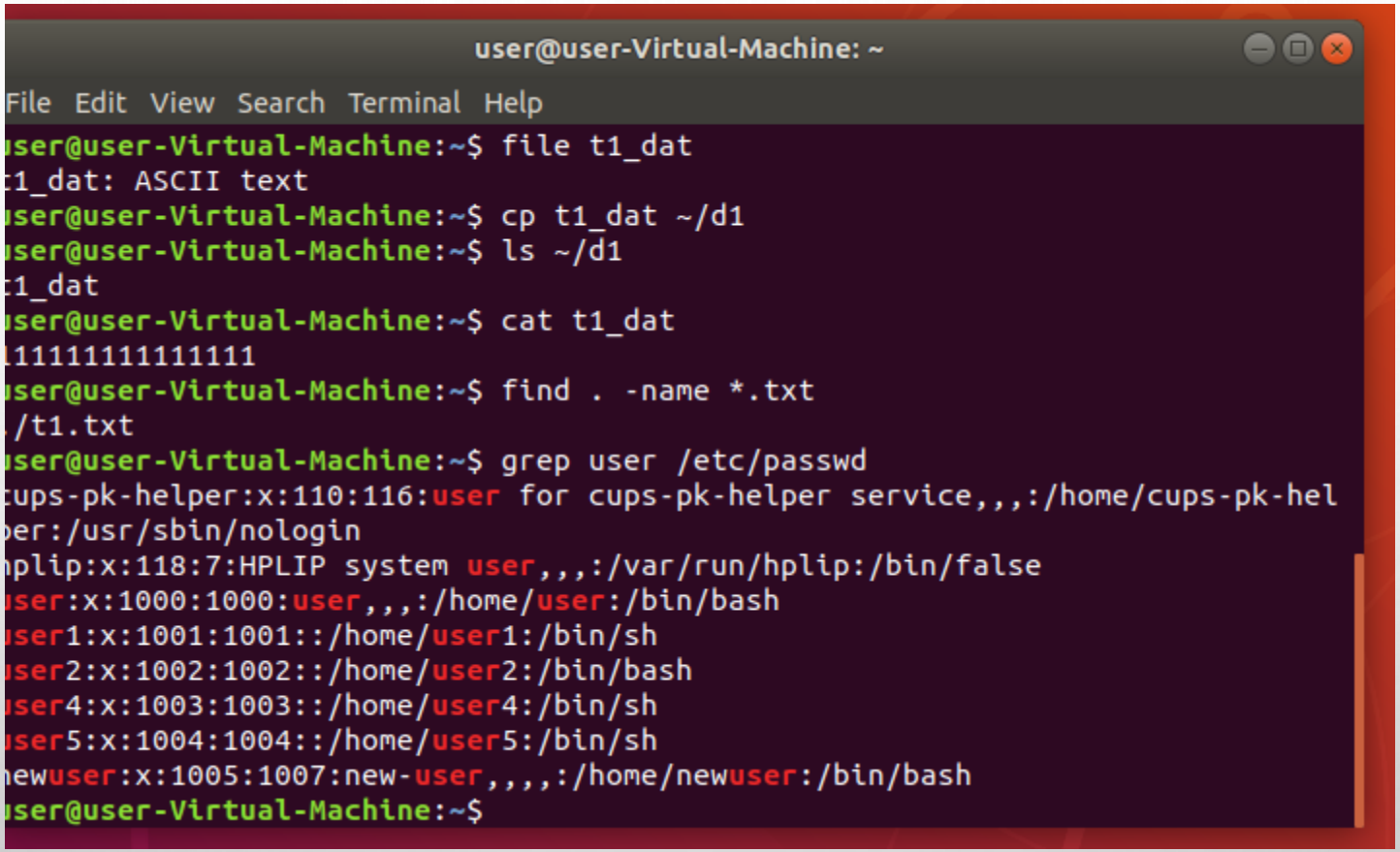
MAIN COMMANDS WORKING WITH DIRECTORIES

COMMAND	DESCRIPTION	EXAMPLE
pwd	To print the current directory	pwd
mkdir	To create a new directory (directories), if it doesn't exist	mkdir ~/d1 ~/d2
cd	To change the current directory (to move to another directory)	cd d1
ls	To print the specified directory content	Ls -il


user@user-Virtual-Machine: ~/d1
File Edit View Search Terminal Help
user@user-Virtual-Machine:~$ man mkdir
user@user-Virtual-Machine:~$ mkdir ~/d1 ~/d2
user@user-Virtual-Machine:~$ ls -l
total 68
drwxr-xr-x 2 user user 4096 мая 19 14:55 d1
drwxr-xr-x 2 user user 4096 мая 19 14:55 d2
drwxr-xr-x 2 user user 4096 фев 21 12:15 Desktop
drwxr-xr-x 2 user user 4096 фев 21 12:15 Documents
drwxr-xr-x 2 user user 4096 фев 21 12:15 Downloads
-rw-r--r-- 1 user user 8980 фев 21 12:13 examples.desktop
drwxr-xr-x 2 user user 4096 фев 21 12:15 Music
lrwxrwxrwx 1 user user 11 мая 19 14:13 my_passwd -> /etc/passwd
drwxr-xr-x 2 user user 4096 мая 16 13:42 Pictures
drwxr-xr-x 2 user user 4096 фев 21 12:15 Public
drwxr-xr-x 0 root root 0 янв 1 1970 shared-drives
drwxr-xr-x 3 user user 4096 апр 28 12:58 snap
-rw-r--r-- 1 user user 16 мая 19 14:10 t1_dat
-rw-r--r-- 1 user user 0 мая 19 14:08 t1.txt
drwxr-xr-x 2 user user 4096 фев 21 12:15 Templates
-rw-rw-r-- 1 user user 7885 мар 29 13:31 'Untitled 1.odt'
drwxr-xr-x 2 user user 4096 фев 21 12:15 Videos
user@user-Virtual-Machine:~$ cd d1
user@user-Virtual-Machine:~/d1$ pwd
/home/user/d1
user@user-Virtual-Machine:~/d1$ ls -il
total 0
user@user-Virtual-Machine:~/d1$ ls -il ..
total 68
1038761 drwxr-xr-x 2 user user 4096 мая 19 14:55 d1
1039156 drwxr-xr-x 2 user user 4096 мая 19 14:55 d2

MAIN COMMANDS TO WORK WITH FILES

COMMAND	DESCRIPTION	EXAMPLE
cat	Displays a “filename”, or concatenate files	cat 1.txt
cp	Copies one file/directory to the specified location	cp 1.txt ~\d1
file	Identifies the file type (binary, text, etc)	file 1.txt
find	Finds a file/directory	find . –iname “*.jpg” –look for files with names according to template “*.jpg” in the current directory ignoring case of symbols
grep	Looks for text in a file	grep User /etc/passwd


user@user-Virtual-Machine: ~
File Edit View Search Terminal Help
user@user-Virtual-Machine:~$ file t1_dat
t1_dat: ASCII text
user@user-Virtual-Machine:~$ cp t1_dat ~/d1
user@user-Virtual-Machine:~$ ls ~/d1
t1_dat
user@user-Virtual-Machine:~$ cat t1_dat
111111111111111
user@user-Virtual-Machine:~$ find . -name *.txt
./t1.txt
user@user-Virtual-Machine:~$ grep user /etc/passwd
cups-pk-helper:x:110:116:user for cups-pk-helper service,,,:/home/cups-pk-hel
per:/usr/sbin/nologin
hplip:x:118:7:HPLIP system user,,,:/var/run/hplip:/bin/false
user:x:1000:1000:user,,,:/home/user:/bin/bash
user1:x:1001:1001::/home/user1:/bin/sh
user2:x:1002:1002::/home/user2:/bin/bash
user4:x:1003:1003::/home/user4:/bin/sh
user5:x:1004:1004::/home/user5:/bin/sh
newuser:x:1005:1007:new-user,,,,:/home/newuser:/bin/bash
user@user-Virtual-Machine:~$

FILE ATTRIBUTES

Number of hard links

The owner

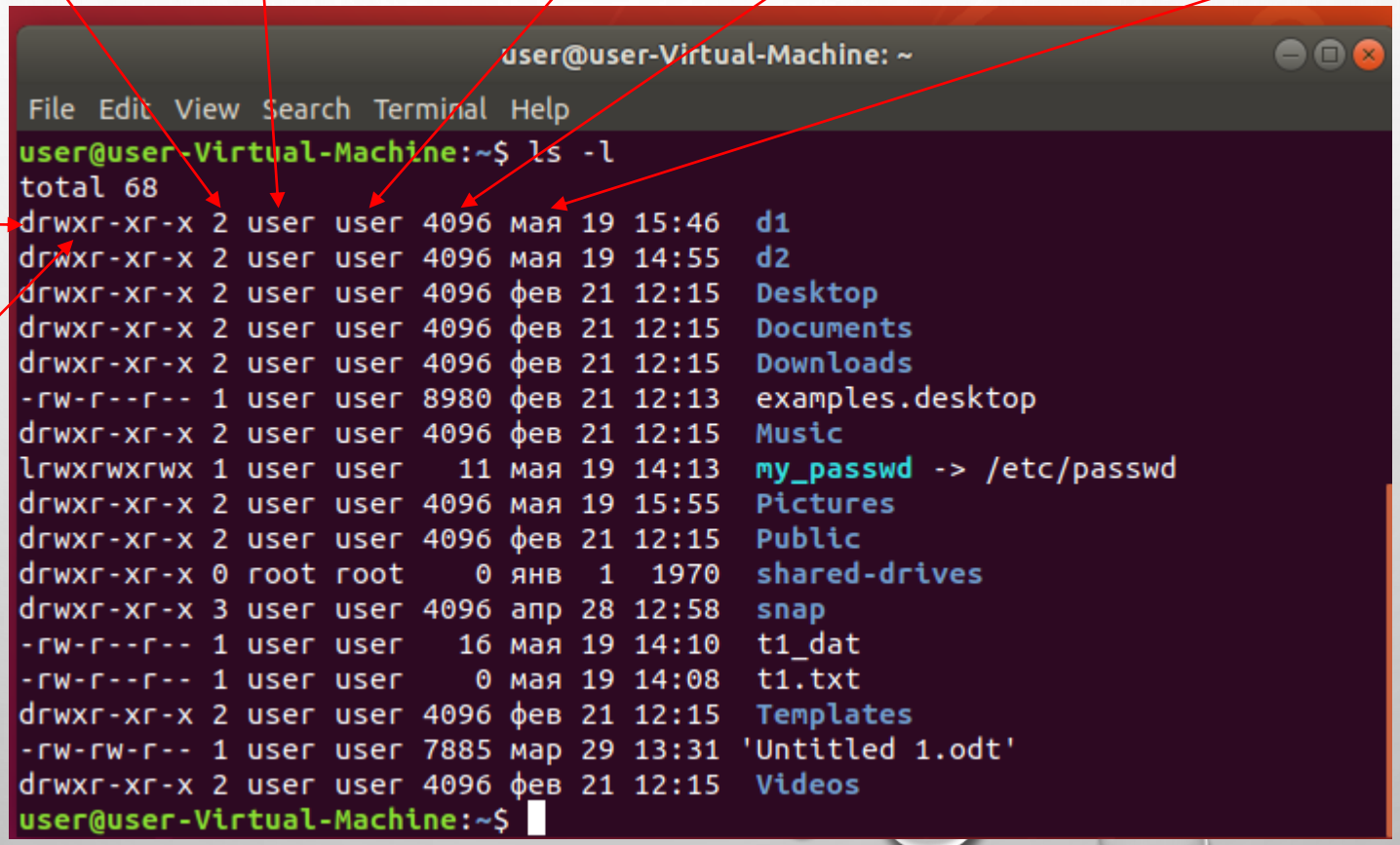
The group owner

File size

Date and time of the last modification

Type of the file

Permissions of access



FILE TYPES (“LS” COMMAND)

- **-** : REGULAR FILE
- **d** : DIRECTORY
- **c** : CHARACTER DEVICE FILE
- **b** : BLOCK DEVICE FILE
- **s** : LOCAL SOCKET FILE
- **p** : NAMED PIPE
- **l** : SOFT LINK

ACCESS PERMISSIONS

- ARE SET FOR:
  - ✓ FOR THE OWNER (A SINGLE USER)
  - ✓ FOR THE GROUP OWNER
  - ✓ FOR THE OTHERS
- THREE STANDARD PERMISSIONS (RIGHTS):
  - ✓ READ
  - ✓ WRITE
  - ✓ EXECUTE

STANDARD PERMISSIONS

- READ
  - ✓ LOOK AT THE CONTENT OF A FILE
  - ✓ FIND OUT WHICH FILES ARE IN A DIRECTORY
- WRITE
  - ✓ CHANGE OR DELETE THE CONTENT OF A FILE
  - ✓ CREATE OR DELETE FILES FROM A DIRECTORY
- EXECUTE
  - ✓ CAN EXECUTE (RUN AS A PROGRAM)
  - ✓ CAN WORK WITH THE FILES IN A DIRECTORY

CHANGING FILE PERMISSIONS (CHMOD IN SYMBOLIC SYNTAX))

THE BASIC FORMAT OF THE COMMAND IS:

chmod MODE FILENAME

WHERE:

MODE (IN SYMBOLIC SYNTAX): WHO OPERATIONS PERMISSIONS

WHO: u -> FILE OWNER

g -> GROUP OWNER

o-> OTHERS

a -> ALL

OPERATIONS: + ->ADD PERMISSIONS

- ->REMOVE PERMISSIONS

= ->SET PERMISSIONS

PERMISSIONS: R- -> READ

W->WRITE

X ->EXECUTE

EXAMPLE: chmod a+x file_name

CHANGING FILE PERMISSIONS (CHMOD IN OCTAL FORMAT)

OCTAL FORMAT THE MODE IS BASED UPON A OCTAL NUMBER REPRESENTING THE DIFFERENT MODE PERMISSIONS, WHERE EACH OF THE PERMISSION GROUPS (USER, GROUP, OTHERS) HAS AN OCTAL VALUE REPRESENTING THE READ, WRITE AND EXECUTE BITS.

THE MAIN BENEFITS OF USING OCTAL FORMAT IS THAT ALL THE PERMISSIONS ARE SET AT THE SAME TIME AND THE COMMAND IS MUCH SHORTER THAN IF ALL THE PERMISSIONS WERE SET USING THE SYMBOLIC FORMAT.

CHMOD (OCTAL FORMAT)

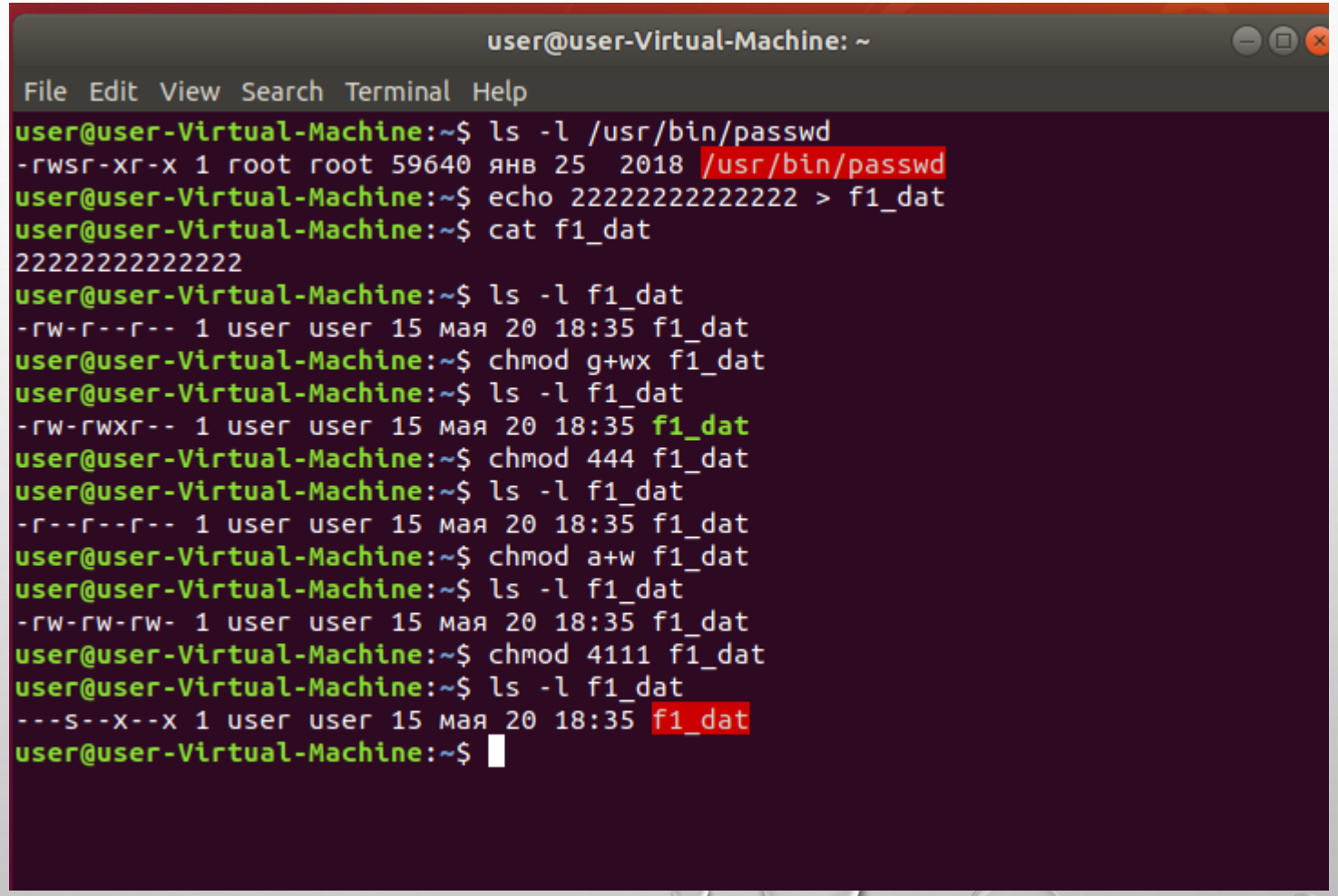
	USER	GROUP	OTHERS
SYMBOLIC	rwx	rw-	r--
BINARY	111	110	100
	4+2+1	4+2+0	4+0+0
OCTAL	7	6	4

EXAMPLE: chmod 764 file_name

SPECIAL PERMISSIONS (SUID, SGID, STICKY BIT)

- SUID (USED ON AN EXECUTABLE FILE) THIS WILL MAKE THE COMMAND RUN AS THE OWNER OF THE FILE (OFTEN ROOT) RATHER THAN THE PERSON THAT LAUNCHED THE APPLICATION.
  - ✓ THIS IS REQUIRED WHERE A REGULAR USER NEEDS TO RUN A PROGRAM THAT NEEDS TO ACCESS FILES THAT CAN ONLY BE READ/WRITTEN TO BY ROOT.
  - ✓ FOR EXAMPLE THE PASSWD COMMAND WHICH NEEDS TO ACCESS THE PROTECTED SHADOW PASSWORD FILE IS SUPPORTED BY FILE /USR/BIN/PASSWD (SEE NEXT SLIDE)
- THE SETGID BIT IS SIMILAR TO THE SETUID BIT, BUT IS SET ON THE GROUP PERMISSIONS.
- THE STICKY BIT IS USED TO RESTRICT WHO CAN DELETE A FILE IN A DIRECTORY. WHEN THIS IS SET ON A DIRECTORY THEN ONLY THE OWNER OF A FILE (OR ROOT) CAN DELETE A FILE WITHIN THAT DIRECTORY.
- TO SET ANY OF THESE SPECIAL PERMISSIONS USING THE OCTAL FORMAT THEN AN ADDITIONAL DIGIT IS USED BEFORE THE REST OF THE OCTAL NUMBER (4-SUID, 2-SGID, 1-STICKY BIT)

EXAMPLES


user@user-Virtual-Machine: ~
File Edit View Search Terminal Help
user@user-Virtual-Machine:~$ ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 59640 янв 25 2018 /usr/bin/passwd
user@user-Virtual-Machine:~$ echo 22222222222222 > f1_dat
user@user-Virtual-Machine:~$ cat f1_dat
22222222222222
user@user-Virtual-Machine:~$ ls -l f1_dat
-rw-r--r-- 1 user user 15 мая 20 18:35 f1_dat
user@user-Virtual-Machine:~$ chmod g+wx f1_dat
user@user-Virtual-Machine:~$ ls -l f1_dat
-rw-rwxr-- 1 user user 15 мая 20 18:35 f1_dat
user@user-Virtual-Machine:~$ chmod 444 f1_dat
user@user-Virtual-Machine:~$ ls -l f1_dat
-r--r--r-- 1 user user 15 мая 20 18:35 f1_dat
user@user-Virtual-Machine:~$ chmod a+w f1_dat
user@user-Virtual-Machine:~$ ls -l f1_dat
-rw-rw-rw- 1 user user 15 мая 20 18:35 f1_dat
user@user-Virtual-Machine:~$ chmod 4111 f1_dat
user@user-Virtual-Machine:~$ ls -l f1_dat
---s--x--x 1 user user 15 мая 20 18:35 f1_dat
user@user-Virtual-Machine:~$

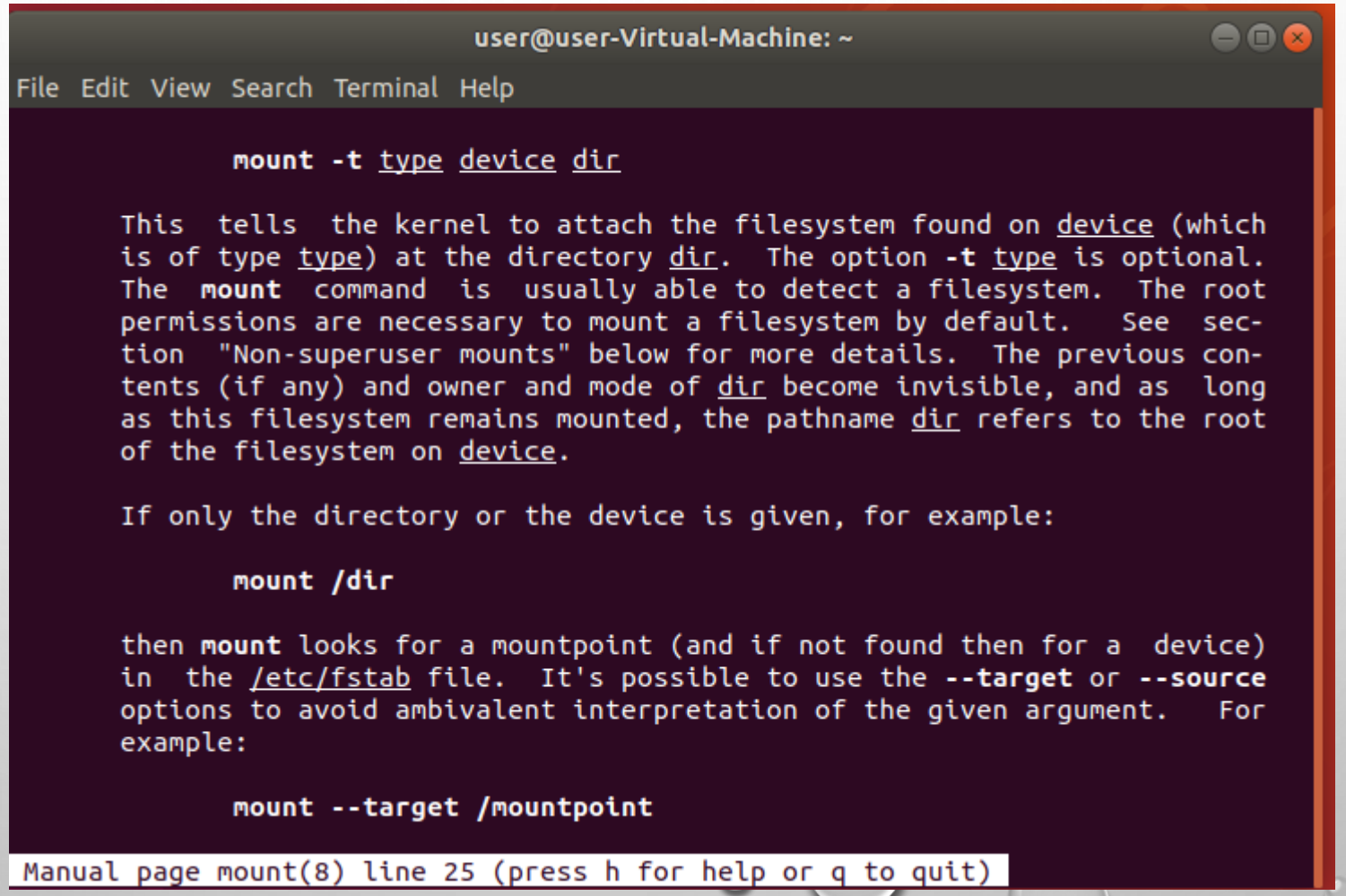
“MKFS” COMMAND

- **MKFS** IS USED TO BUILD A LINUX FILE SYSTEM ON A DEVICE, USUALLY A HARD DISK PARTITION.
- SYNTAX: mkfs [options] [-t fs-type] device [size]
- Where:
  - ✓ -t fs-type -> SPECIFIES THE TYPE OF FILE SYSTEM TO BE BUILT. IF NOT SPECIFIED, THE DEFAULT FILE SYSTEM TYPE (CURRENTLY **EXT2**) IS USED;
  - ✓ device -> HARD DISK PARTITION FILE (/dev/hda1, dev/sdb2, …)
- Example: mkfs –t ext2 /dev/sda5

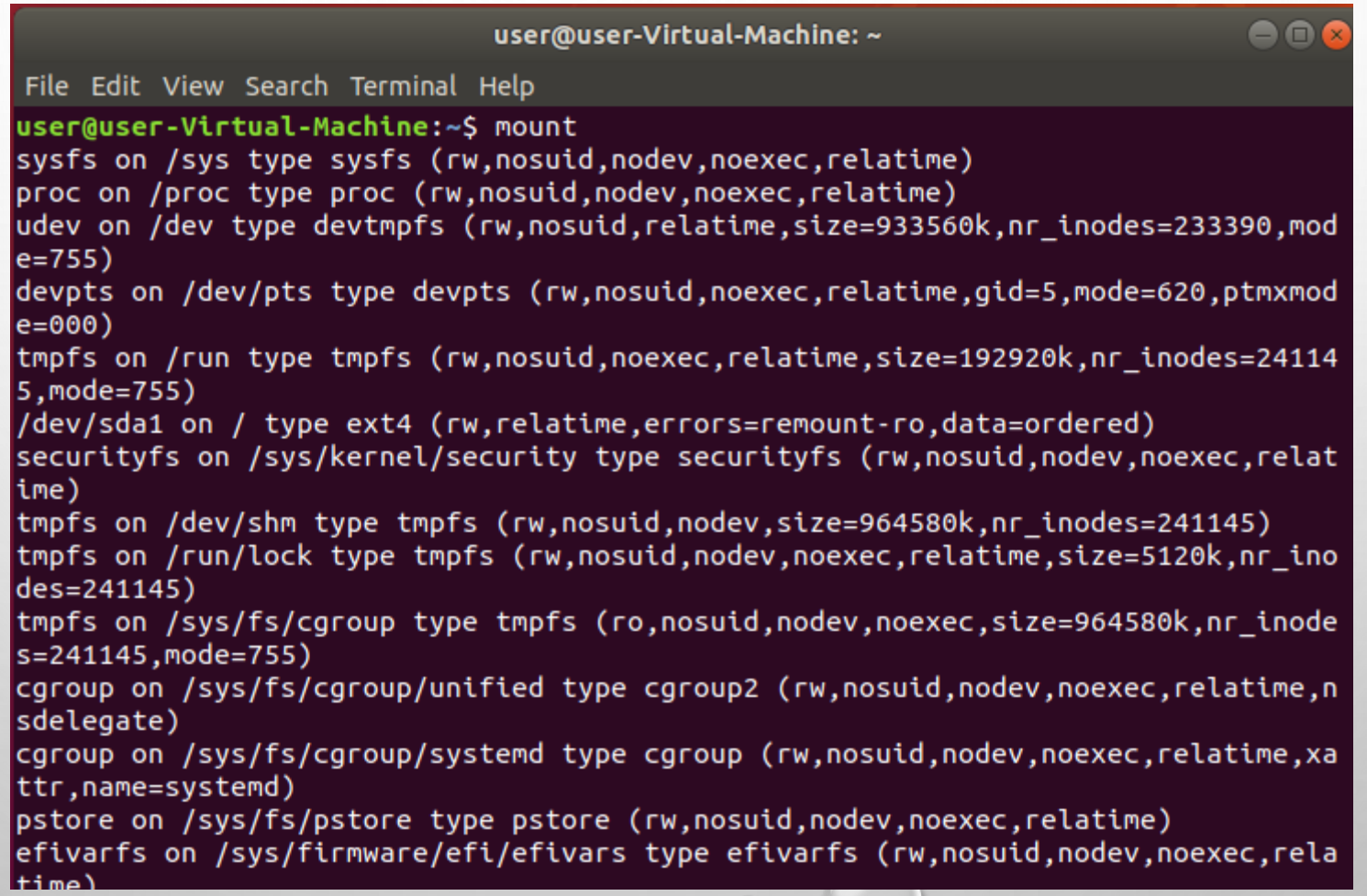
MOUNT / UMOUNT

- THE **MOUNT** COMMAND MOUNTS A STORAGE DEVICE OR FILESYSTEM, MAKING IT ACCESSIBLE AND ATTACHING IT TO AN EXISTING DIRECTORY STRUCTURE.
- ALL FILES ACCESSIBLE IN LINUX ARE ARRANGED IN ONE BIG TREE: THE FILE HIERARCHY, ROOTED AT /. THESE FILES CAN BE SPREAD OUT OVER SEVERAL DEVICES. THE **MOUNT** COMMAND ATTACHES A FILESYSTEM, LOCATED ON SOME DEVICE OR OTHER, TO THE FILE TREE.
- THE **UMOUNT** COMMAND "UNMOUNTS" A MOUNTED FILESYSTEM, INFORMING THE SYSTEM TO COMPLETE ANY PENDING READ OR WRITE OPERATIONS, AND SAFELY DETACHING IT.

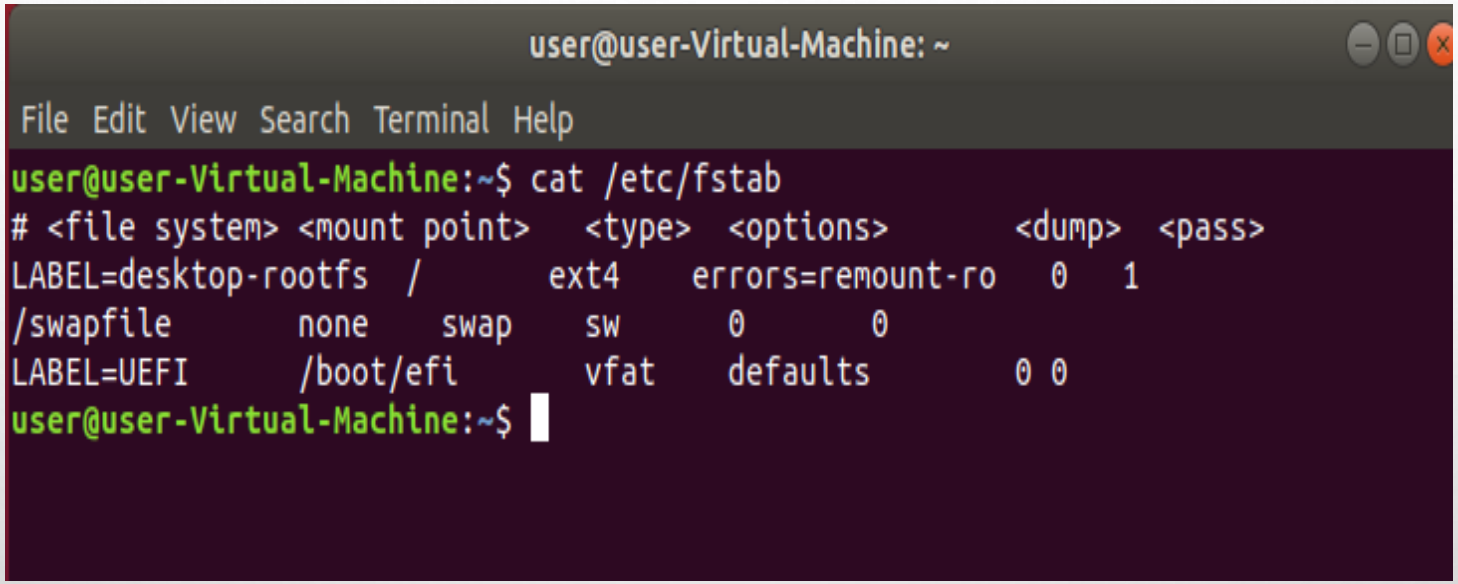
MOUNT SYNTAX



EXAMPLE : MOUNT



FILE “/etc/fstab”



FILE “/etc/fstab”

```
# device-spec   mount-point     fs-type      options                                    dump pass
LABEL=/         /               ext4         defaults                                      1 1
/dev/sda6       none            swap         defaults                                      0 0
none            /dev/pts        devpts       gid=5,mode=620                                0 0
none            /proc           proc         defaults                                      0 0
none            /dev/shm        tmpfs        defaults                                      0 0

# Removable media
/dev/cdrom      /mnt/cdrom      udf,iso9660  noauto,owner,ro                               0 0

# NTFS Windows 7 partition
/dev/sda1       /mnt/Windows    ntfs-3g      quiet,defaults,locale=en_US.utf8,umask=0,noexec  0 0

# Partition shared by Windows and Linux
/dev/sda7       /mnt/shared     vfat         umask=000                                     0 0

# mounting tmpfs
tmpfs           /mnt/tmpfschk   tmpfs        size=100m                                     0 0

# mounting cifs
//pingu/ashare  /store/pingu    cifs         credentials=/root/smbpass.txt                 0 0

# mounting NFS
pingu:/store    /store          nfs          rw                                            0 0
```

“/etc/fstab” STRUCTURE

1.device-spec – The device name, label, or ID, or other means of specifying the partition or data source this entry refers to.
2.mount-point – Where the contents of the device may be accessed after mounting; for swap partitions or files, this is set to none.
3.fs-type – The type of file system to be mounted.
4.options – Options describing various other aspects of the file system, such as whether it is automatically mounted at boot, which users may mount or access it, whether it may be written to or only read from, its size, and so forth; the special option defaults refers to a predetermined set of options depending on the file system type.
5.dump – A number indicating whether and how often the file system should be backed up by the dump program; a zero indicates the file system will never be automatically backed up.
6.pass – A number indicating the order in which the fsck program will check the devices for errors at boot time; this is 1 for the root file system and either 2 (meaning check after root) or 0 (do not check) for all other devices.

HOME TASK

- ✓ HOW MANY PROCESSES ARE RUNNING IN YOUR SYSTEM?
- ✓ WHAT INFORMATION ABOUT EVERY PROCESS IS DELIVERED BY “top” TOOL, AND WHAT IS THE MAIN DIFFERENCE BETWEEN “ps” COMMAND AND “top’’ TOOL?
- ✓ CREATE IN YOUR HOME DIRECTORY A HARD LINK AND A SYMLINK TO ONE OF THE DIRECTORIES FROM THE FIRST LEVEL
- ✓ WHAT WILL BE DONE WITH THE HELP OF THE COMMAND:

```
find ./test1 ./test2 -type f -name "*.py"
```

- ✓ GIVE THE EXAMPLES OF EVERY COMMAND FROM THE TABLES (SLIDES 25, 27)
- ✓ SUPPOSE THAT IN YOUR CURRENT DIRECTORY THERE IS AN EXECUTABLE FILE NAMED “EXAMPLE1”.
- ✓ WRITE THE COMMAND TO CHANGE ACCESS PERMISSIONS AND TO GIVE ALL PERMISSIONS TO ITS OWNER, PERMISSION TO READ ANDV EXECUTE TO THE GROUP OWNER, AND ONLY PERMISSION TO EXECUTE TO THE OTHERS.