



# LINUX BASICS

Internship  
Spring 2025

# Meet the speaker



**Igor Bannicov**

Senior DevOps Consultant @ Endava



# Agenda

- 01 Operating systems
- 02 Linux system architecture
- 03 Linux file system
- 04 Package managers
- 05 Processes in Linux
- 06 User management
- 07 Terminal/ssh/scp
- 08 Command for learning commands
- 09 Moving through file tree
- 10 Working with files
- 11 Working with file content



# AGENDA

---

- OPERATING SYSTEMS
- LINUX SYSTEM ARCHITECTURE
- LINUX FILE SYSTEM
- PACKAGE MANAGERS

PROCESSES IN LINUX

USER MANAGEMENT

TERMINAL/SSH/SCP

COMMAND FOR LEARNING COMMANDS

MOVING THROUGH FILE TREE

WORKING WITH FILES

WORKING WITH FILE CONTENT

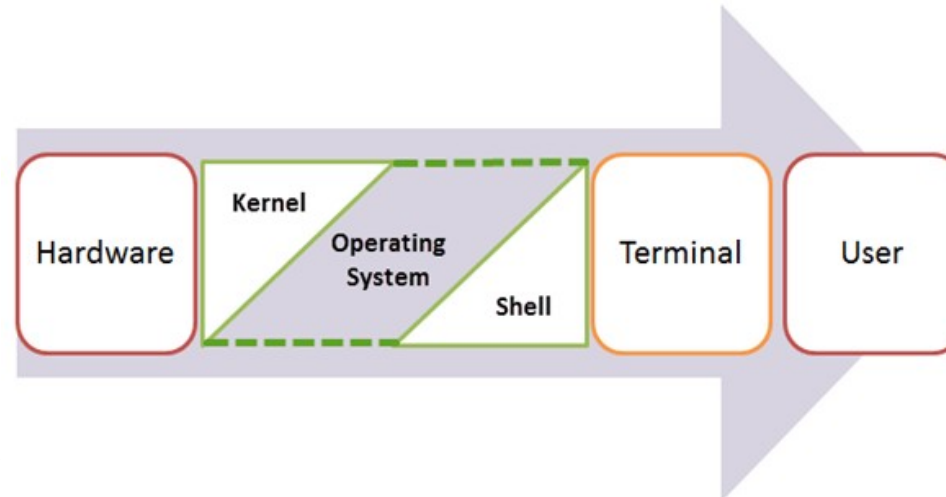
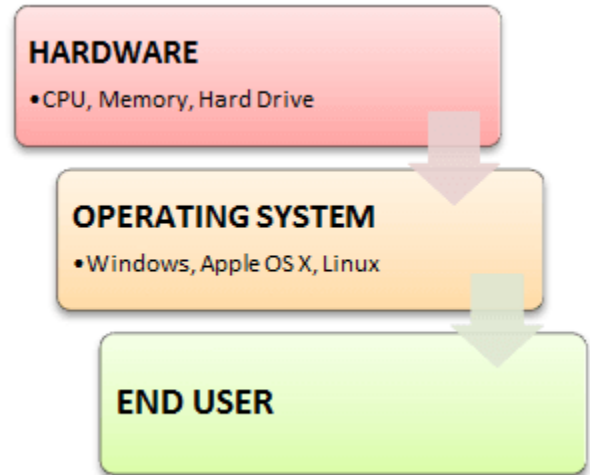


# OPERATING SYSTEMS

---

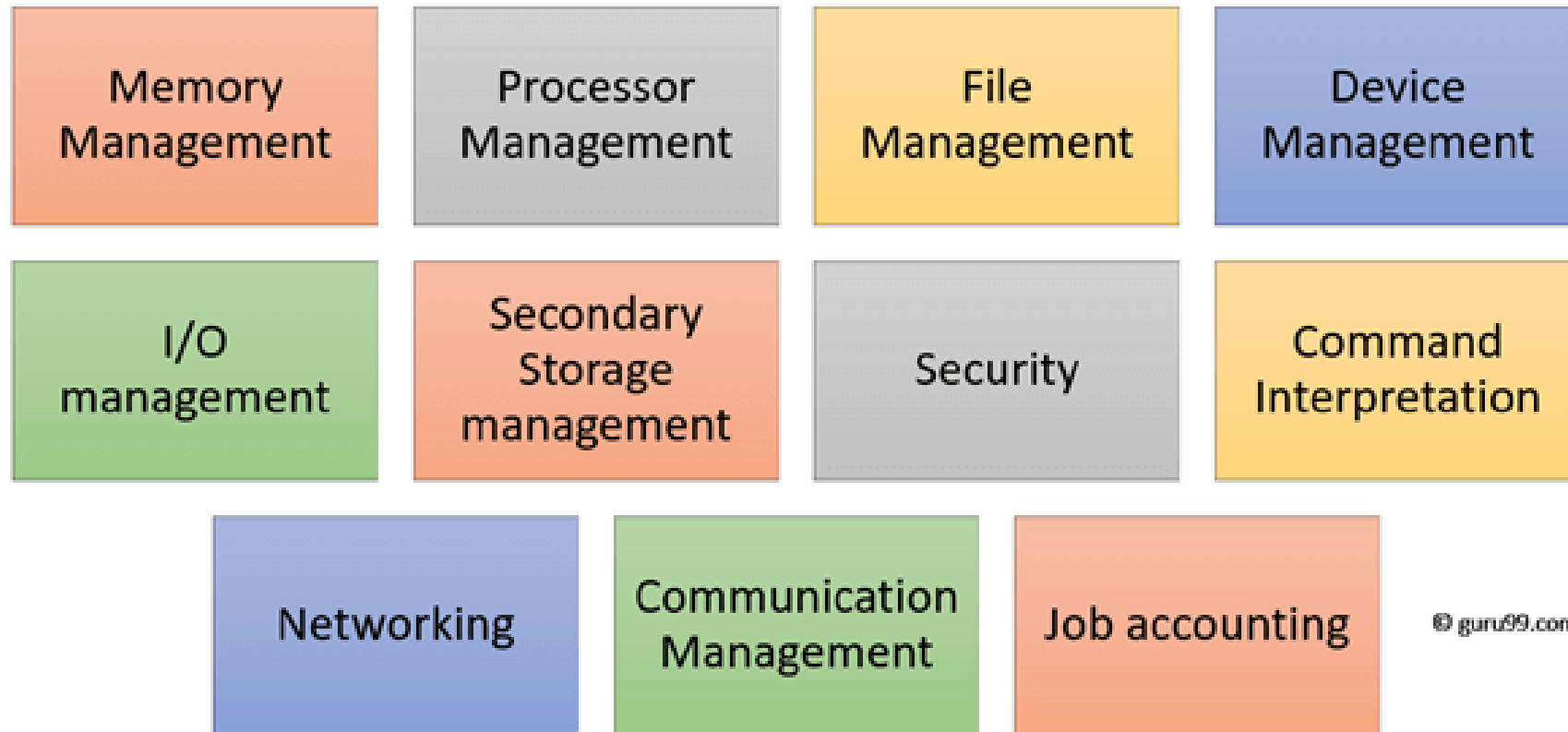
## FEATURES OF OPERATING SYSTEM

- PROTECTED AND SUPERVISOR MODE
- ALLOWS DISK ACCESS AND FILE SYSTEMS DEVICE DRIVERS NETWORKING SECURITY
- PROGRAM EXECUTION
- MEMORY MANAGEMENT VIRTUAL MEMORY MULTITASKING
- HANDLING I/O OPERATIONS
- MANIPULATION OF THE FILE SYSTEM
- ERROR DETECTION AND HANDLING
- RESOURCE ALLOCATION
- INFORMATION AND RESOURCE PROTECTION



# FUNCTIONS OF AN OPERATING SYSTEM

.....

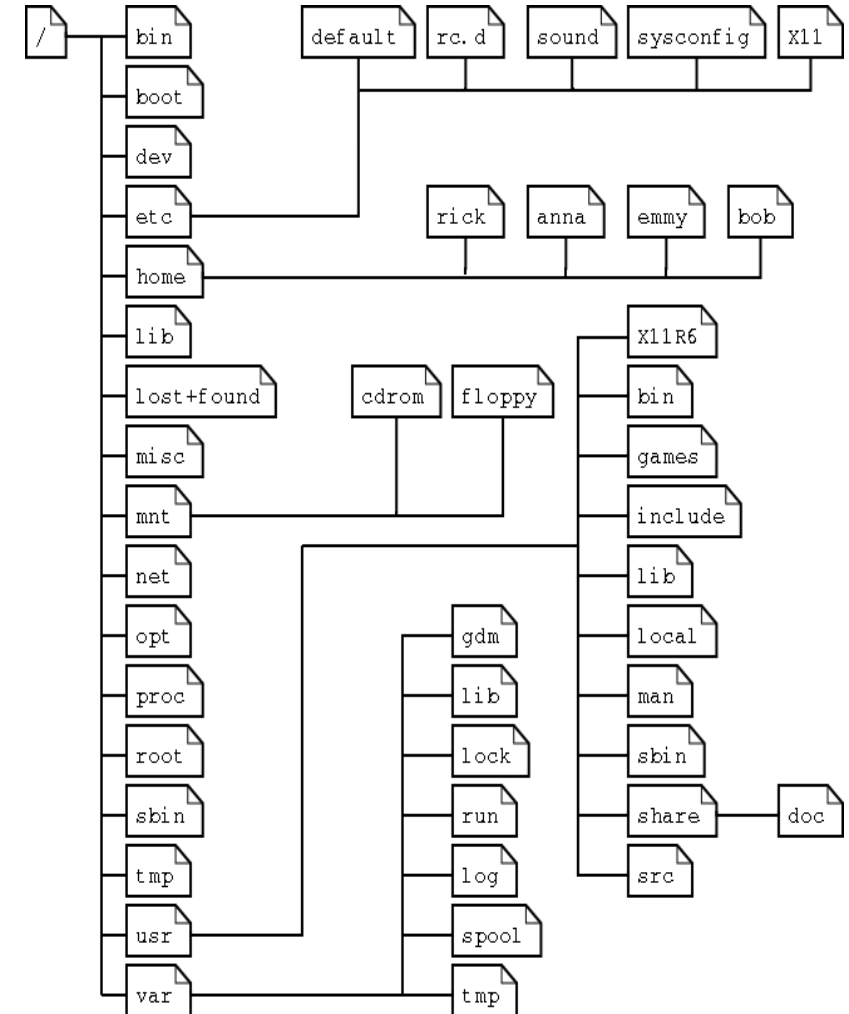


© guru99.com

# LINUX FILE SYSTEM

## MAIN FOLDERS

- / - root folder for fs
- /boot – bootable files
- /bin – common programs
- /dev – references to the devices
- /etc - configs
- /home – users home folders
- /lib – kernel modules
- /mnt – standard mount point
- /proc – system resources info
- /opt – 3<sup>rd</sup> party and extra software
- /root – home folder for root user
- /run - processes
- /usr – user`s program
- /var – variables and temporary data
- /sys – system information
- /sbin – system binaries



# PACKAGE MANAGERS

---

## PACKAGE TYPES

### Package tools:

**rpm** – Debian, Mint, Ubuntu  
**dpkg/deb** – APT, Synaptic

### Package managers:

**yum** - RedHat, Fedora, SUSE  
**apt** - Debian, Ubuntu  
**pacman, yaourt** - Arch

**Source-based** – Slackware, Gentoo

## WORKING WITH PACKAGES

`apt-get update` -

`apt-get upgrade`

`apt-get install <app name>`

`apt-cache search <package name>`

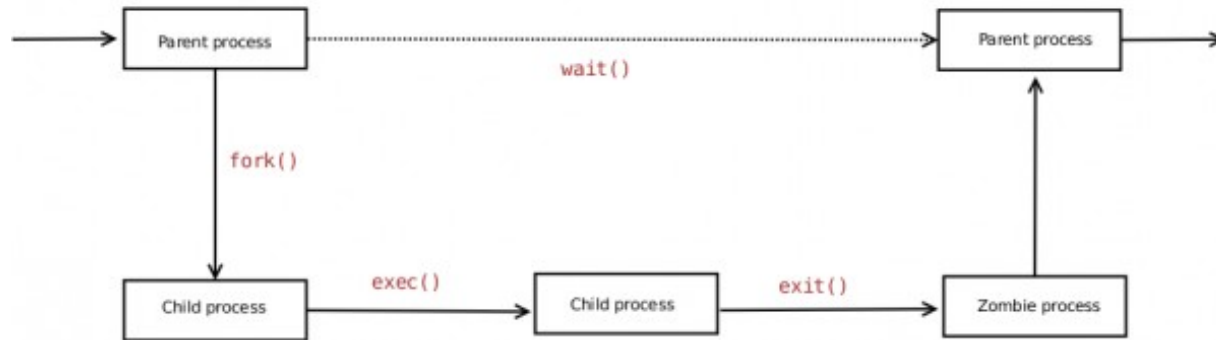
`apt-cache showpkg <package name>`





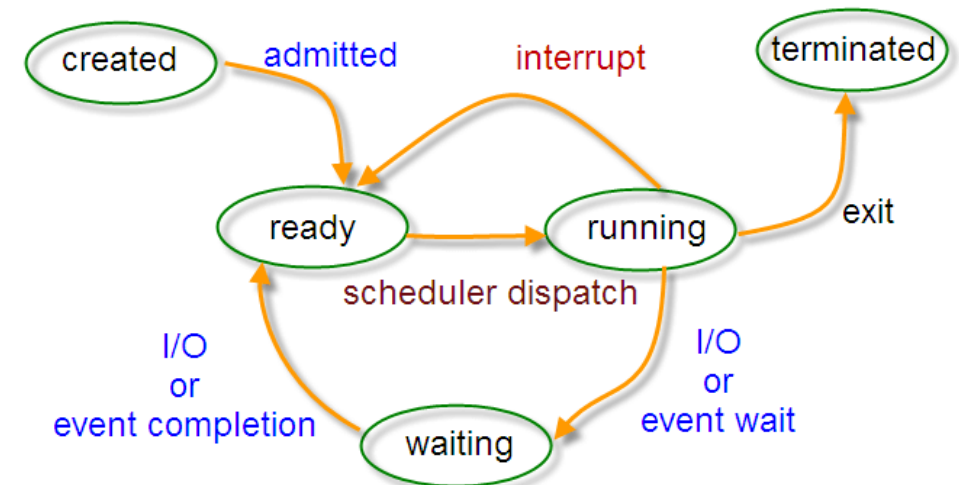
# PROCESSES

Linux Processes Life Cycle



- **Running** – it is the current process in the system.
- **Waiting** – in this state, a process is waiting for an event to occur or for a system resource.
- **Stopped** – in this state, a process has been stopped, usually by receiving a signal.
- **Zombie** – here, a process is dead, it has been halted but it's still has an entry in the process table.

## Process State



# USER MANAGEMENT

---

**R**EAD =100

**W**RITE =010

**E**XECUTE=001

Command	Description
<code>sudo adduser username</code>	Adds a user
<code>sudo passwd -l 'username'</code>	Disable a user
<code>sudo userdel -r 'username'</code>	Delete a user
<code>sudo usermod -a -G GROUPNAME USERNAME</code>	Add user a to a usergroup
<code>sudo deluser USER GROUPNAME</code>	Remove user from a user group
<code>whoami</code>	prints your user name
<code>chown &lt;username&gt;:&lt;groupname&gt;</code>	Change file owner and group

# COMMAND FOR LEARNING COMMANDS

---

- **MAN**, I DON'T KNOW A COMMAND <ABC>

- JUST ASK **MAN**, DUDE

**man** == manual pages

**man** <command> - manual about <command>

**man** <config.file> - manual about <config.file>

**man** <daemon> - manual about <daemon>

## I DON'T WANNA **READ TO MUCH**

**whatis** – shows the first line of manual

**whereis** – the location of manual page

**info** – just like man but in the info format

# MOVING THROUGH FILE TREE

---

## LETS WALK

`pwd` – Print Working Directory

`cd` – Change Directory

`ls` - LiSt

`mkdir` – MaKe DIRectory

## I'M JUST TO TIRED TO TYPE ALL OF THESE LETTERS

Use Tab key for autocomplete !!!!!

And about absolute and relative paths

And remember shortcuts!!

`/.../.../...` - Absolute path

`.` – current directory

`.../.../...` – Relative path.

`..` – a parent directory

`~` - home directory

# WORKING WITH FILES

---

## REMEMBER NEXT RULES

Case sensitivity !

Everything is a file !

## SO WHAT CAN WE DO WITH FILES?!

`file` – determines the file type

`touch` – create an empty file

`rm` – ReMove file

`cp` – CoPy file

`mv` – MoVe. Or Rename))

`rename` – Rename multiple files according to the regexp

# WORKING WITH FILE CONTENT

---

## OK. SO WHAT`S NEXT ?

**head** – show the first 10 lines of file

**tail** – show the last 10 lines of file

**cat** – an universal tool to work with files

**tac** – an opposite for cat command

**more / less** – prints file on screen

**strings** – prints readable strings from (binary) files

**nano** – simple text editor

**emacs** – powerful and customizable text editor

**vi / vim** – most powerful text editor you`ve ever seen ;)

# PROCESSES IN LINUX

## WHAT IS USING MY CPU ?!

**ps** – Process Status

**ps -ef / ps aux** – see every process

**ps axjf** – build a process tree

**ps o pid,user,command**

**top/htop/atop** – list of all working processes

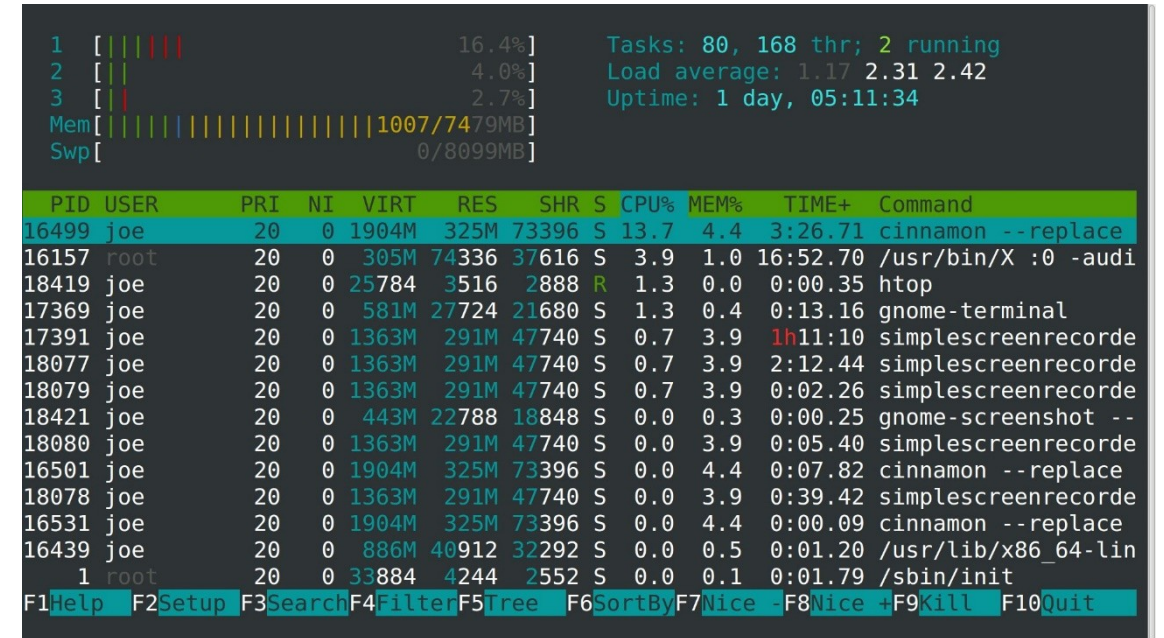
## WELL... I NEED TO KILL SOMETHING

**kill** – kill the process

**kill <pid>** - try to kill a process

**kill -9 <pid>** - kill a process with SIGTERM

**kill -<signal> <pid>**



The screenshot shows the htop interface. At the top, system statistics are displayed: 1 task running, 2 tasks in the running state, 4.0% CPU usage, 2.7% memory usage, 1007/7479MB memory used, and 0/8099MB swap used. Below this, a table of running processes is shown. The table has columns for PID, USER, PRI, NI, VIRT, RES, SHR, S, CPU%, MEM%, TIME+, and Command. The processes listed include cinnamon --replace, /usr/bin/X :0 -audi, htop, gnome-terminal, simplescreenrecorde, gnome-screenshot --, and /sbin/init.

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
16499	joe	20	0	1904M	325M	73396	S	13.7	4.4	3:26.71	cinnamon --replace
16157	root	20	0	305M	74336	37616	S	3.9	1.0	16:52.70	/usr/bin/X :0 -audi
18419	joe	20	0	25784	3516	2888	R	1.3	0.0	0:00.35	htop
17369	joe	20	0	581M	27724	21680	S	1.3	0.4	0:13.16	gnome-terminal
17391	joe	20	0	1363M	291M	47740	S	0.7	3.9	1h11:10	simplescreenrecorde
18077	joe	20	0	1363M	291M	47740	S	0.7	3.9	2:12.44	simplescreenrecorde
18079	joe	20	0	1363M	291M	47740	S	0.7	3.9	0:02.26	simplescreenrecorde
18421	joe	20	0	443M	22788	18848	S	0.0	0.3	0:00.25	gnome-screenshot --
18080	joe	20	0	1363M	291M	47740	S	0.0	3.9	0:05.40	simplescreenrecorde
16501	joe	20	0	1904M	325M	73396	S	0.0	4.4	0:07.82	cinnamon --replace
18078	joe	20	0	1363M	291M	47740	S	0.0	3.9	0:39.42	simplescreenrecorde
16531	joe	20	0	1904M	325M	73396	S	0.0	4.4	0:00.09	cinnamon --replace
16439	joe	20	0	886M	40912	32292	S	0.0	0.5	0:01.20	/usr/lib/x86_64-lin
1	root	20	0	33884	4244	2552	S	0.0	0.1	0:01.79	/sbin/init

# SSH/SCP

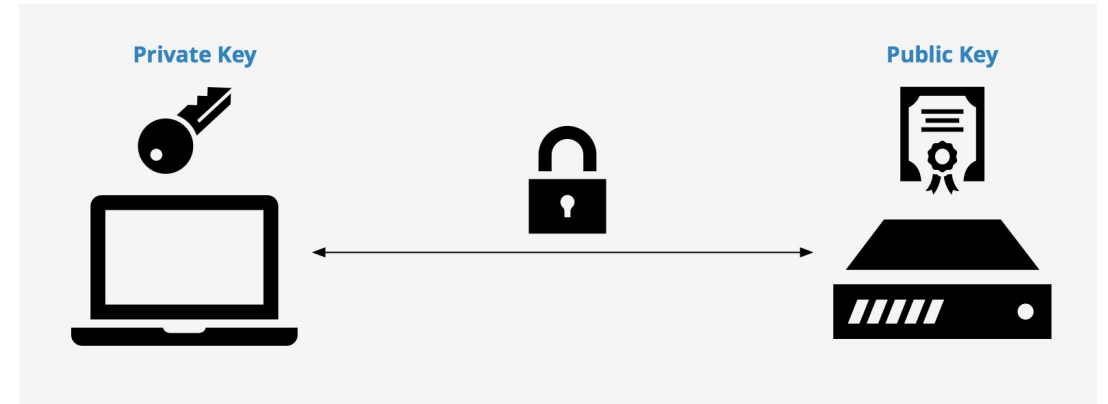
---

## WHAT ABOUT REMOTES?

`ssh` = Secure SHell

`ssh-keygen` – generate a ssh key

`ssh -p <port> <ip>`



## HOW CAN I SEND FILES WITHOUT GUI ?

`scp` = Secure CoPy

`scp <remote_host>:<file> <local destination>`

`scp <local source> <remote host>:<remote destination>`

`scp <remote host1>:<remote dest1> <remote host 2>:<remote dest2>`







# Linux

**DIONIS MOLDOVAN**  
SENIOR DEVOPS ENGINEER