



Press here

#LEARN IN DEPTH

#Be professional in
embedded system

1

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

Embedded Linux (PART 5)

- BASICS LINUX COMMANDS
- USING WILD CARDS
- LINUX HELP COMMANDS
- COMPOSITE COMMANDS
- SHELL SCRIPT
- SEARCHING TEXT (GREP COMMAND)
- COMPARING TEXT FILES
- GIT

ENG.KEROLES SHENOUDA

Eng. Keroles Shenouda

Embedded Linux

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Eng.keroles.karam@gmail.com



#LEARN IN DEPTH
#Be professional in
embedded system

eng. Keroles Shenouda
<https://www.facebook.com/groups/embedded.system.KS/>

<https://www.learn-in-depth.com/>

<https://www.facebook.com/groups/embedded.system.KS/>

2

Basics Linux commands



#LEARN IN DEPTH
#Be professional in
embedded system

3

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

Files & Filenames

- ▶ Filenames can use any characters
 - ▶ abc??##.a1 sta*.pn
 - ▶ But it is not wise to use special characters in filenames
- ▶ spaces are accepted but not recommended, use dashes or underscores instead
 - ▶ test results.txt => test_results.txt or test-results.txt
- ▶ File names are case sensitive
 - ▶ Test_results.doc test_results.doc
 - ▶ But avoid creating files with same names with difference in case
- ▶ File names starting with a dot are hidden files
 - ▶ .bashrc .profile
- ▶ No concept of file extension, the dot is just another character
 - ▶ file.txt.old .bashrc file.doc.mod .profile.old results.yesterday results

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN IN DEPTH
#Be professional in
embedded system

4

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

Paths

- ▶ Directory names separated by slashes '/'
 - ▶ /usr/src/shared/files/myfile.txt
- ▶ Can be absolute or relative
 - ▶ Absolute: Does not depend on where you are
 - ▶ ~/labs == /home/user_name/labs/

```
embedded_system_ks@embedded-KS:~/labs$ pwd
/home/embedded_system_ks/labs
. . . . .
```

- ▶ Relative: Depends where you are in the tree
 - ▶ ./my-project/progress-reports
 - ▶ ../../my-project/progress-reports
 - ▶ . means Current Directory
 - ▶ .. Means Parent Directory

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



5

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>#LEARN IN DEPTH
#Be professional in
embedded system

Basic Commands

Command	Effect
\$ ls	To list files
\$ tree	To list a tree of files
\$ pwd	To show the current directory
\$ cd	To move around in the tree
\$ mkdir	To Create directories
\$ cp	To copy files and directories
\$ mv	To move or rename files and directories
\$ rm	To delete files and directories
\$ clear	To clear the screen

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



6

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

Listing files (ls)

- ▶ If you want to see the list of files on your UNIX or Linux system, use the 'ls' command.
 - ▶ Note:
 - ▶ Directories are denoted in blue color.
 - ▶ Files are denoted in white.
- ▶ You can use 'ls -R' to shows all the files not only in directories but also subdirectories

```
guru99@VirtualBox:~$ ls
Desktop    Downloads      Music      Public     Videos
Documents  examples.desktop Pictures  Templates
guru99@VirtualBox:~$
```

```
guru99@VirtualBox:~$ ls -R
.:
Desktop    Downloads      Music      Public     Videos
Documents  examples.desktop Pictures  Templates

./Desktop:
./Documents:
./Downloads:
./Music:
./Music/English:
./Music/English/Rock:
./Music/English/Trans:
./Music/English/Test.mp3

./Pictures:
./Public:
./Templates:
./Videos:
guru99@VirtualBox:~$
```

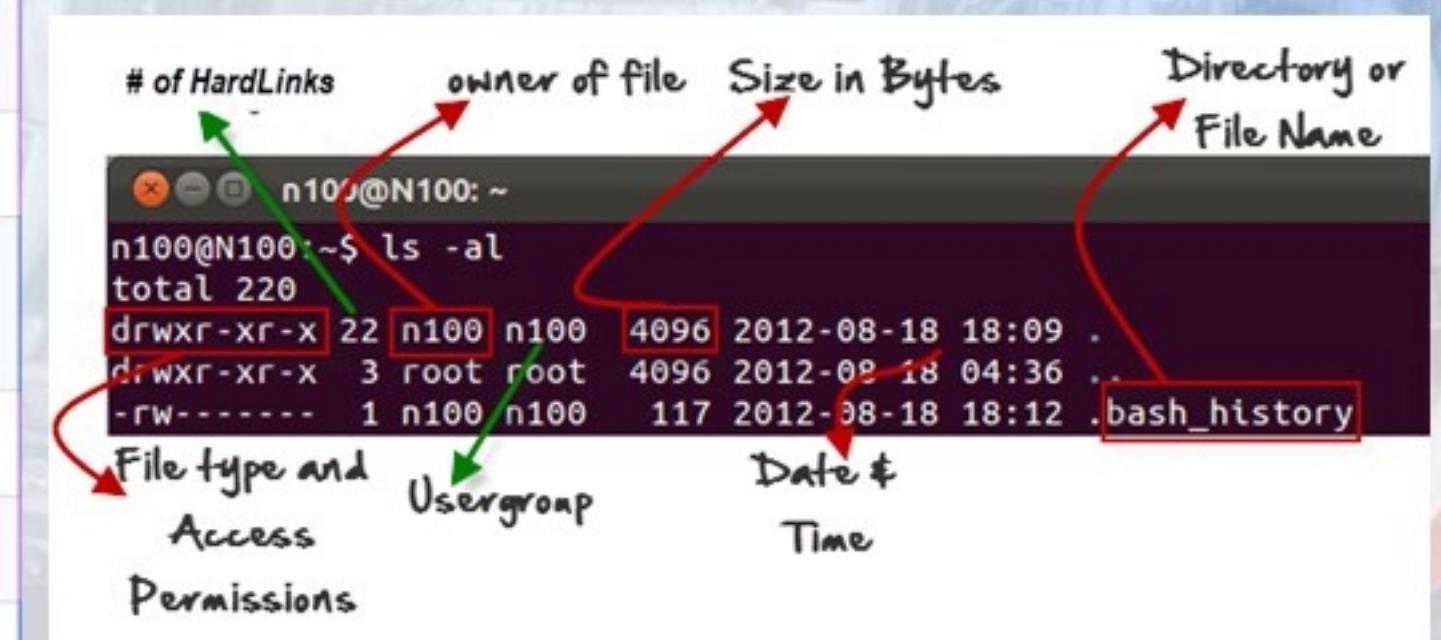


<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Listing files (ls)

- 'ls -al' gives detailed information of the files. The command provides information in a columnar format. The columns contain the following information:

1 st Column	File type and access permissions
2 nd Column	# of HardLinks to the File
3 rd Column	Owner and the creator of the file
4 th Column	Group of the owner
5 th Column	File size in Bytes
6 th Column	Date and Time
7 th Column	Directory or File name



The screenshot shows the terminal command 'ls -al' being run, resulting in a detailed listing of files and directories. The listing includes columns for file type, permissions, hard links, owner, group, size, date, and time, followed by the file name. Handwritten annotations with arrows point from the table columns to these specific fields in the terminal output.

# of HardLinks	owner of file	Size in Bytes	Directory or File Name
22	n100 n100	4096	.
3	root root	4096	..
1	n100 n100	117	.bash_history

Annotations:

- 1st Column: Points to the 'File type and Access Permissions' column in the terminal output.
- 2nd Column: Points to the '# of HardLinks' column in the terminal output.
- 3rd Column: Points to the 'Usergroup' column in the terminal output.
- 4th Column: Points to the 'Date & Time' column in the terminal output.
- 5th Column: Points to the 'Size in Bytes' column in the terminal output.
- 6th Column: Points to the 'File type and Access Permissions' column in the terminal output.
- 7th Column: Points to the 'Directory or File Name' column in the terminal output.



<https://www.facebook.com/groups/embedded.system.KS/>

eng. Keroles Shenouda

#LEARN IN DEPTH
#Be professional in
embedded system

8

(ls Command)

\$ ls [options] [<dir or file> ..]

- \$ ls** (list current directory)
- \$ ls -a** (all: show hidden files)
- \$ ls -l** (long: show file details)
- \$ ls -t** (timestamp: sort based on timestamp)
- \$ ls -S** (Size: sort based on file size)
- \$ ls -r** (reverse: make the sort in reverse order)
- \$ ls -d** (directories: Only show directories)
- \$ ls -R** (Recursive: list files inside subdirectories)
- \$ ls <dir>** (List the contents of the mentioned directory)
- \$ ls <file>** (List the mentioned)
- \$ ls <dir or file> <dir or file> <dir or file>** (list selected dirs or files)

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN IN DEPTH
#Be professional in
embedded system

9

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

Creating & Viewing Files

- ▶ The 'cat' command is used to display text files. It can also be used for copying, combining and creating new text files. Let's see how it works.

- ▶ To create a new file, use the command
 - ▶ cat > filename
 - ▶ Add content
 - ▶ Press 'ctrl + d' to return to command prompt.

- ▶ To view a file, use the command -
 - ▶ cat filename

```
guru99@VirtualBox:~$ cat sample1
This is sample1
```

Create a File
guru99@VirtualBox:~\$ cat > sample1
Enter Content
This is sample1
Press Control + D to exit
guru99@VirtualBox:~\$



#LEARN IN DEPTH
#Be professional in
embedded system

10

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

Creating & Viewing Files

- ▶ The syntax to combine 2 files is -
 - ▶ `cat file1 file2 > newfilename`

```
guru99@VirtualBox:~$ cat sample1 sample2 > sample
```

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN IN DEPTH
#Be professional in
embedded system

11

eng. Keroles Shenouda

<https://www>

Deleting Files

- ▶ The 'rm' command removes files from the system without confirmation.
- ▶ To remove a file use syntax -
 - ▶ rm filename
- ▶ To remove directory
 - ▶ Rm -r

List current contents of directory

```
guru99@VirtualBox:~$ ls
Desktop   Downloads   Music   Public  sample1  Templates
Documents examples.desktop Pictures sample  sample2  Videos
```

Remove the file sample1

```
guru99@VirtualBox:~$ rm sample1
```

List directory , to check file has been deleted

```
guru99@VirtualBox:~$ ls
Desktop   Downloads   Music   Public  sample2  Videos
Documents examples.desktop Pictures sample  Templates
guru99@VirtualBox:~$
```

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Moving and Re-naming files

- ▶ To move or rename a file, use the command.
 - ▶ `mv filename new_file_location`
- ▶ For renaming file:
 - ▶ `mv filename newfilename`

```
guru99@VirtualBox:~$ mv test test1
guru99@VirtualBox:~$ ls
Desktop  Downloads  Music  Public  test1
Documents examples.desktop  Pictures  Templates  Videos
guru99@VirtualBox:~$ █
```

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN IN DEPTH

#Be professional in
embedded system

13

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

Creating Directories

- ▶ Directories can be created on a Linux operating system using the following command
 - ▶ `mkdir dirname`

```
home@VirtualBox:~$ mkdir mydirectory
home@VirtualBox:~$ ls
Desktop  Downloads      Music      Pictures  Templates
Documents examples.desktop mydirectory  Public    Videos
home@VirtualBox:~$
```

▶ Renaming Directory

- ▶ The 'mv' (move) command (covered earlier) can also be used for renaming directories. Use the below-given format:

```
home@VirtualBox:~$ mv mydirectory newdirectory
home@VirtualBox:~$ ls
Desktop  Downloads      Music      Pictures  Templates
Documents examples.desktop newdirectory  Public    Videos
home@VirtualBox:~$
```

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN IN DEPTH

#Be professional in
embedded system

14

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

The 'Man' command

- ▶ To get help on any command that you do not understand, you can type
 - ▶ **man**

```
guru99@VirtualBox:~$ man man

guru99@VirtualBox:~$ man(1)          Manual pager utils          MAN(1)
NAME
       man - an interface to the on-line reference manuals

SYNOPSIS
       man [-C file] [-d] [--warnings[=warnings]] [-R encoding] [-L
       locale] [-m system[,...]] [-M path] [-s list] [-e extension] [-i|-I]
       [--regex|--wildcard] [--names-only] [-a] [-u] [--no-subpages] [-P
       pager] [-r prompt] [-7] [-E encoding] [--no-hyphenation] [--no-justifi-
       cation] [-p string] [-t] [-T[device]] [-H[browser]] [-X[dpi]] [-Z]
       [[section] page ...]
       man -k [apropos options] regexp ...
       man -K [-w|-W] [-S list] [-i|-I] [--regex] [section] term ...
       man -f [whatis options] page ...
       man -l [-C file] [-d] [--warnings[=warnings]] [-R encoding] [-L
       locale] [-P pager] [-r prompt] [-7] [-E encoding] [-p string] [-t]
       [-T[device]] [-H[browser]] [-X[dpi]] [-Z] file ...
       man -w|-W [-C file] [-d] [-D] page ...
       man -c [-C file] [-d] [-D] page ...
       man [-hv]

DESCRIPTION
       Manual page man(1) line 1 (press h for help or q to quit)
```

<https://www.learn-in-depth.com/>

<https://www.facebook.com/groups/embedded.system.KS/>



<https://www.facebook.com/groups/embedded.system.KS/>

eng. Keroles Shenouda

#LEARN IN DEPTH
#Be professional in
embedded system

15

The History Command

- ▶ History command shows all the commands that you have used in the past for the current terminal session. This can help you refer to the old commands you have entered and re-used them in your operations again.

```
guru99@VirtualBox:~$ history
1 cat > sample
2 cat sample
3 cat sample ^a
4 cat sample a
5 cat sample | grep a
6 cat sample | grep ^a
7 useradd home
8 useradd mycomputer
9 sudo useradd mycomputer
10 sudo adduser MyLinux
11 sudo adduser mylinux
12 vi scriptsample.sh
```

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN IN DEPTH

#Be professional in
embedded system

16

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

The clear command

- ▶ This command clears all the clutter on the terminal and gives you a clean window to work on, just like when you launch the terminal.

```
141 man
142 3a
143 man intro
144 man ls
145 man cat
146 man man
147 history
148 146
149 history 146
150 history
151 clear
152 history
guru99@VirtualBox:~$ clear
```

The window gets cleared

```
guru99@VirtualBox:~$
```

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Displaying the Directory Tree (tree Command)

- ▶ \$ tree (List tree from Current Directory)

```
bashitout@mike-desktop:~/projects$ tree
.
└── awesomesite
    ├── awesomesite
    │   ├── __init__.py
    │   ├── settings.py
    │   ├── urls.py
    │   └── wsgi.py
    └── manage.py

2 directories, 5 files
bashitout@mike-desktop:~/projects$
```

- ▶ \$ tree -a (List tree from Current Directory)
- ▶ \$ tree [options] [.dir or file.]
 - ▶ \$ tree (display the full tree starting from current dir)
 - ▶ \$ tree -d (only show directories)
 - ▶ \$ tree -a (show all files; including hidden ones)

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Print Working Directory (pwd Command)

- ▶ \$ pwd
- ▶ \$ pwd (Display Current Directory)

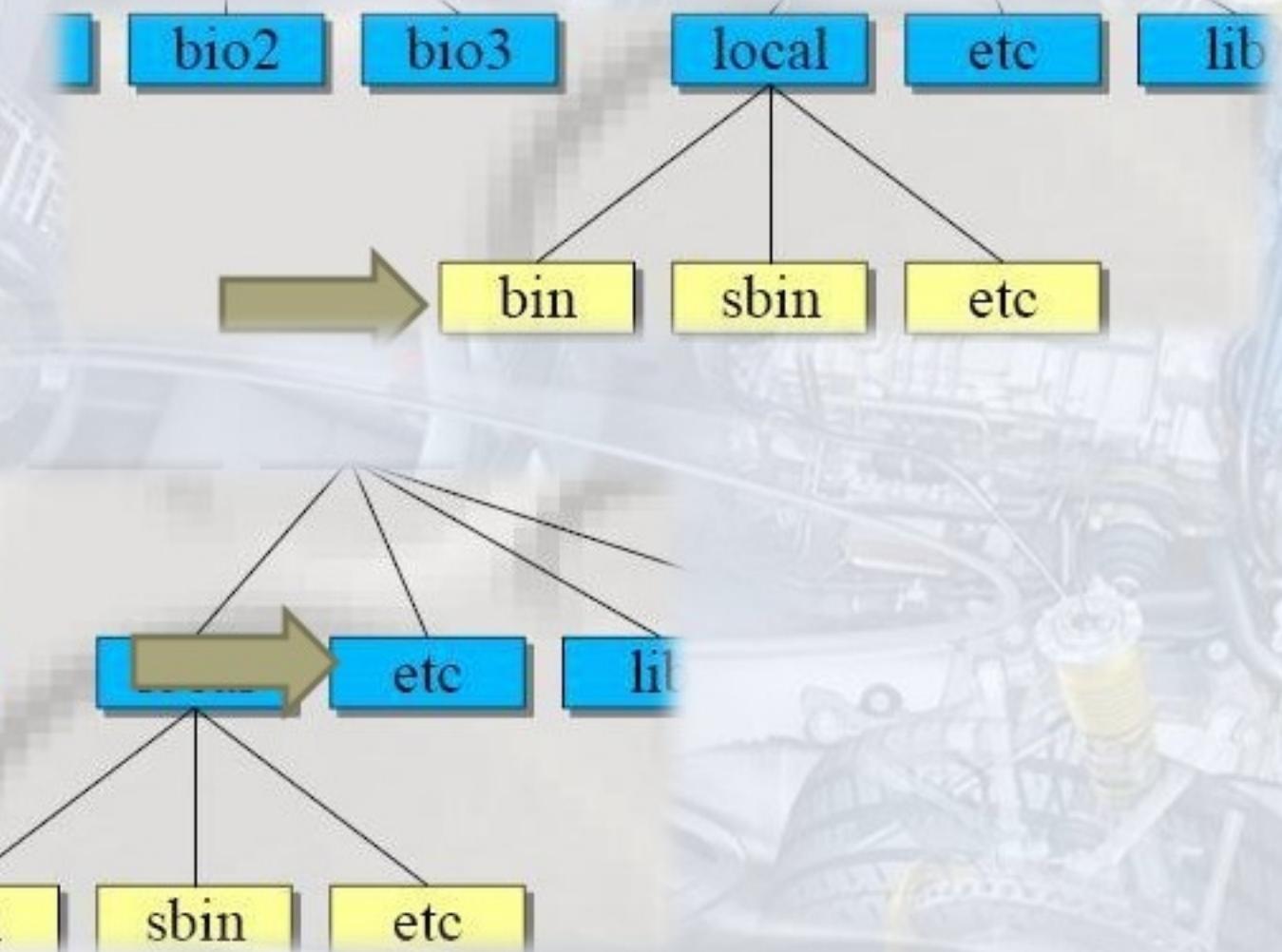
```
vivek@wks01:~/scripts$ pwd
/home/vivek/scripts
vivek@wks01:~/scripts$ ls -l
total 32
-rwxr-xr-x 1 vivek vivek 545 Sep 28 2008 conditions.sh
-rw-r--r-- 1 vivek vivek 1720 Sep 28 2008 gmail_parser.py
-rwxr-xr-x 1 vivek vivek 56 Dec 1 2007 hddmonit.sh
drwxr-xr-x 2 vivek vivek 4096 Mar 10 01:24 php
-rwxr-xr-x 1 vivek vivek 10283 Jul 20 2012 pogodynka.sh
drwxr-xr-x 2 vivek vivek 4096 Mar 10 01:24 python
vivek@wks01:~/scripts$ cd python
vivek@wks01:~/scripts/python$ pwd
/home/vivek/scripts/python
vivek@wks01:~/scripts/python$ █
```

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



Moving Around (cd Command)

- ▶ \$ cd [destination]
- ▶ \$ cd /usr/local/bin



▶ \$ cd /usr/etc

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



<https://www.facebook.com/groups/embedded.system.KS/>

eng. Keroles Shenouda

#LEARN IN DEPTH
#Be professional in
embedded system

20

(cd Command)

\$ cd [destination]

\$ cd /etc/network (absolute path)

\$ cd ../project/ (relative path)

\$ cd ./project (relative path)

\$ cd project (relative path, same as **./project**)

\$ cd ~ (go to my home directory **/home/aelarabawy/**)

\$ cd ~user_name (go to **/home/user_name**)

\$ cd (same as **cd ~**)

\$ cd .. (go to parent directory)

\$ cd - (go to previous directory)

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Display a Line of Text

- ▶ (echo Command)
- ▶ \$echo <text>
- ▶ \$echo \$<variable_name>

```
embedded_system_ks@embedded-KS: ~
embedded_system_ks@embedded-KS:~$ echo "learn-in-depth"
learn-in-depth
embedded_system_ks@embedded-KS:~$ echo "learn-in-depth" > log
log2.text  log.text
embedded_system_ks@embedded-KS:~$ echo "learn-in-depth" > log.text
embedded_system_ks@embedded-KS:~$ more log.text
learn-in-depth
embedded_system_ks@embedded-KS:~$ cat log.text
learn-in-depth
embedded_system_ks@embedded-KS:~$ export variable_1="hello"
embedded_system_ks@embedded-KS:~$ echo $variable_1
hello
embedded_system_ks@embedded-KS:~$
```

Embedded Linux

ENG.KEROLES SHENOUDA

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Showing the Date and Time (date Command)

- ▶ \$ date (Show current date and time)
 - ▶ Is very useful to get time-stamps

```
embedded_system_ks@embedded-KS:~  
embedded_system_ks@embedded-KS:~$ date  
الجمعة 27 ديسمبر 23:40:31 EET 2019  
embedded_system_ks@embedded-KS:~$ date  
الجمعة 27 ديسمبر 23:40:33 EET 2019  
embedded_system_ks@embedded-KS:~$ date +%D  
12/27/19  
embedded_system_ks@embedded-KS:~$ date +%F  
2019-12-27  
embedded_system_ks@embedded-KS:~$ date +%j  
361  
embedded_system_ks@embedded-KS:~$ date +%Y  
2019  
embedded_system_ks@embedded-KS:~$ date +%m/%d/%Y  
12/27/2019  
embedded_system_ks@embedded-KS:~$ date +%m-%d-%Y  
12-27-2019  
embedded_system_ks@embedded-KS:~$
```

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN IN DEPTH
#Be professional in
embedded system

23

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

Basic Calculator

- ▶ (bc Command)
 - ▶ The bc tool provides a basic math calculator
 - ▶ It has capabilities from basic math, to assigning variables, to dealing with arrays

```
embedded_system_ks@embedded-KS:~$ bc
bc 1.06.95
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
1+2*4
9
12+1
13
quit
embedded_system_ks@embedded-KS:~$ echo "12+5" | bc
17
embedded_system_ks@embedded-KS:~$ echo "10^2" | bc
100
embedded_system_ks@embedded-KS:~$ x=`echo "12+5" | bc` 
embedded_system_ks@embedded-KS:~$ echo $x
17
embedded_system_ks@embedded-KS:~$ echo "var=10;++var" | bc
11
embedded_system_ks@embedded-KS:~$ echo "10>5" | bc
1
embedded_system_ks@embedded-KS:~$ echo "10<5" | bc
0
embedded_system_ks@embedded-KS:~$ echo 'n=8;m=10;if(n>m) print "n is greater" else print "m is greater"' | bc -l
n is greater
embedded_system_ks@embedded-KS:~$
```

Variable is increased first and then result of variable is stored.

Conditional Statements are used to take decisions and execute statements based on these decisions. bc command supports the if condition.

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Printing System Information

- ▶ (uname Command)

```
embedded_system_ks@embedded-KS: ~
embedded_system_ks@embedded-KS:~$ uname -a
Linux embedded-KS 4.10.0-42-generic #46-Ubuntu SMP Mon Dec 4 14:36:05 UTC 2017 i686 i686 i686 GNU/Linux
embedded_system_ks@embedded-KS:~$ █
```

Embedded Linux

ENG.KEROLES SHENOUDA

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Reboot the System (reboot Command)

- ▶ \$ reboot (Reboot the system)
 - ▶ This command allows the user (must be root) to reboot the system

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



<https://www.facebook.com/groups/embedded.system.KS/>

eng. Keroles Shenouda

#LEARN IN DEPTH
#Be professional in
embedded system

26

Using Wild Cards

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN IN DEPTH

#Be professional in
embedded system

27

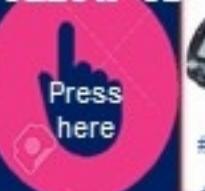
eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

Wild Cards

- ▶ Sometimes you will need to execute a command on a group of files instead of a single file
 - ▶ Examples
 - ▶ You want to delete all log files
 - ▶ You want to list all image files
 - ▶ You want to copy old files (ending with .old) to a different place
 - ▶ Wild cards are patterns that work as placeholders in file names and directory names that are used to apply the command on a group of files/directories that share something in their name

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



The “*” Wild Card

- ▶ The “*” can replace any set of characters (including none) in the file/directory name

▶ \$ rm *.php

```
scott@zen:~/web_site$ ls
400.shtml      experience.php
401.shtml      favicon.ico
403.shtml      fsgb1589.txt
404.shtml      google79b49b8d6cd3bd54.html
500.php        hireme.php
500.shtml      images
about_me.php   includes
archive       index.php
articles      logos
blogger.html   mediawiki2dokuwiki.php
contact.php    ok.htm
credits.html   php_tutorial
css           portfolio.php
docs          privacy.php
scott@zen:~/web_site$
```

```
publications.old
publications.php
samples
sendemail.php
sendMessage.php
services.php
site_credits.php
sitemap.html
speaking.php
style.css
techdocs
wmd-edit.html
```

The “*” Wild Card

- ▶ The “*” can replace any set of characters (including none) in the file/directory name

▶ \$ rm p*

```
scott@zen:~/web_site$ ls
400.shtml      experience.php
401.shtml      favicon.ico
403.shtml      fsgb1589.txt
404.shtml      google79b49b8d6cd3bd54.html
500.php        hireme.php
500.shtml      images
about_me.php   includes
archive       index.php
articles      logos
blogger.html   mediawiki2dokuwiki.php
contact.php   ok.htm
credits.html  phpTutorial ←
css            portfolio.php
docs           privacy.php
scott@zen:~/web_site$
```

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN IN DEPTH
#Be_professional_in_embedded_system

30

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

The “*” Wild Card

- ▶ The “*” can replace any set of characters (including none) in the file/directory name

▶ \$ rm *.htm*

```
scott@zen:~/web_site$ ls
400.shtml      experience.php
401.shtml      favicon.ico
403.shtml      fsgb1589.txt
404.shtml      google79b49b8d6cd3bd54.html
500.php        hireme.php
500.shtml      images
about_me.php   includes
archive       index.php
articles      logos
blogger.html   mediawiki2dokuwiki.php
contact.php   ok.htm
credits.html  php_tutorial
css           portfolio.php
docs          privacy.php
scott@zen:~/web_site$
```

publications.old
publications.php
samples
sendeail.php
sendmessage.php
services.php
site_credits.php
sitemap.html
speaking.php
style.css
techdocs
wmd-edit.html

The “*” Wild Card

- ▶ The “*” can replace any set of characters (including none) in the file/directory name
- ▶ \$ rm -r *.*

```
scott@zen:~/web_site$ ls
400.shtml      experience.php
401.shtml      favicon.ico
403.shtml      fsgb1589.txt
404.shtml      google79b49b8d6cd3bd54.html
500.php        hireme.php
500.shtml      images
about_me.php   includes
archive       index.php
articles      logos
blogger.html   mediawiki2dokuwiki.php
contact.php    ok.htm
credits.html   php_tutorial
css           portfolio.php
docs          privacy.php
scott@zen:~/web_site$
```

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



<https://www.facebook.com/groups/embedded.system.KS/>

eng. Keroles Shenouda

#LEARN IN DEPTH
#Be professional in
embedded system

32

The “*” Wild Card

- ▶ The “*” can replace any set of characters (including none) in the file/directory name

▶ \$ rm -r *

```
scott@zen:~/web_sites$ ls
400.shtml      experience.php
401.shtml      favicon.ico
403.shtml      fsgb1589.txt
404.shtml      google79b49b8d6cd3bd54.html
500.php        hireme.php
500.shtml      images
about_me.php   includes
archive       index.php
articles       logos
blogger.html   mediawiki2dokuwiki.php
contact.php    ok.htm
credits.html   php_tutorial
css            portfolio.php
docs           privacy.php
scott@zen:~/web_sites$
```

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



<https://www.facebook.com/groups/embedded.system.KS/>

eng. Keroles Shenouda

#LEARN IN DEPTH
#Be professional in
embedded system

33

The “?” Wild Card

- ▶ The “?” wild card stands for any single character
- ▶ \$ rm 40?.shtml

```
scott@zen:~/web_site$ ls
400.shtml      experience.php
401.shtml      favicon.ico
403.shtml      fsgb1589.txt
404.shtml      google79b49b8d6cd3bd54.html
500.php        hireme.php
500.shtml      images
about_me.php   includes
archive       index.php
articles       logos
blogger.html   mediawiki2dokuwiki.php
contact.php    ok.htm
credits.html   php_tutorial
css            portfolio.php
docs           privacy.php
scott@zen:~/web_site$
```

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

The “?” Wild Card

- ▶ The “?” wild card stands for any single character
- ▶ \$ rm ?0?.shtml

```
scott@zen:~/web_site$ ls
400.shtml      experience.php
401.shtml      favicon.ico
403.shtml      fsgb1589.txt
404.shtml      google79b49b8d6cd3bd54.html
500.php        hireme.php
500.shtml      images
about_me.php   includes
archive        index.php
articles       logos
blogger.html   mediawiki2dokuwiki.php
contact.php    ok.htm
credits.html   php_tutorial
css            portfolio.php
docs           privacy.php
scott@zen:~/web_site$
```

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

[<chars>] and [! <chars>]

- ▶ We can have more restriction than the use of "?" by specifying a limited set of options for the character
 - ▶ "[ars]" : Stands for a **Single** character from the list a,r,s
 - ▶ "[!ars]" : Stands for any **Single** character except for the list a,r,s
 - ▶ "[2-5]" : Stands for a **Single** character from the range 2 to 5
 - ▶ "[!2-5]" : Stands for any **Single** character except for the list 2 to 5
 - ▶ "[a-l]" : Stands for a **Single** character from range of 'a' to 'l'
 - ▶ "[!a-l]" : Stands for any **Single** character except for the list a to l
 - ▶ "[a-chk]" : Stands for a,b,c,h,k
- ▶ Examples:
 - ▶ \$ rm -r ab[c-fh-j]
 - ▶ removes the files/folders named abc,abd,abe,abf,abh,abi,abj

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

[[:<Class Name>:]]

- ▶ [[:<Class Name>:]] stands for a single character belonging to the specified class
- ▶ Some of used classes
 - ▶ [[:alnum:]] Alpha Numeric characters (a-z, A-Z, 0-9)
 - ▶ [[:alpha:]] Alphabets (a-z, A-Z)
 - ▶ [[:digit:]] Digits (0-9)
 - ▶ [[:lower:]] Lower case character (a-z)
 - ▶ [[:upper:]] Upper case character (A-Z)
- ▶ Examples:
 - ▶ \$ cp results-[[:digit:]][[:digit:]]-[[:alpha:]].log ~/log

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Use of Curly Brackets “{ }”

- ▶ Curly brackets are used to group selections
- ▶ Examples
 - ▶ \$ rm {*.log, *.txt}
 - ▶ \$ cp {*.pdf, *.doc} ~/documents/

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Examples of Escape Sequence

Real File Name	File Name representation
my?file.log	my\?file.log
my*file.log	my*file.log
my file (today).txt	my\ file\ \\\(today\\).txt
“my file”	\”my\ file\”
abc[!2]	abc\[!\2\]

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN IN DEPTH
#Be professional in
embedded system

39

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

Linux Help Commands

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

5 Methods to Get Quick Help on Linux Commands

- ▶ 1. Using apropos to search man pages
- ▶ 2. Read man page of the command
- ▶ 3. Display Single Line Description About a Unix Command
- ▶ 4. Use -h or -help option of the command itself
- ▶ 5. Read Info Documents using Unix info Command

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



1. Using apropos to search man pages

▶ Use apropos to search man pages for available Unix commands on a specific functionality.

▶ \$apropos COMMAND or Description

▶ apropos searches a set of database files containing short descriptions of system commands for keywords and displays the result on the standard output.

```
embedded_system_ks@embedded-KS:~$ apropos grep
bzgrep (1)           - search possibly bzip2 compressed files for a regular expression
bzfgrep (1)          - search possibly bzip2 compressed files for a regular expression
bzgrep (1)           - search possibly bzip2 compressed files for a regular expression
egrep (1)            - print lines matching a pattern
fgrep (1)            - print lines matching a pattern
git-grep (1)          - Print lines matching a pattern
grep (1)              - print lines matching a pattern
grep-aptavail (1)    - grep Debian control files
grep-available (1)   - grep Debian control files
grep-dctrl (1)        - grep Debian control files
grep-debtags (1)     - grep Debian control files
grep-status (1)       - grep Debian control files
grepdiff (1)          - show files modified by a diff containing a regex
grepjar (1)           - search files in a jar file for a pattern
lzegrep (1)           - search compressed files for a regular expression
lzfgrep (1)            - search compressed files for a regular expression
lzgrep (1)             - search compressed files for a regular expression
msggrep (1)            - pattern matching on message catalog
pgrep (1)              - look up or signal processes based on name and other attributes
ptargrep (1)           - Apply pattern matching to the contents of files in a tar archive
rgrep (1)              - print lines matching a pattern

xzegrep (1)           - search compressed files for a regular expression
xzfgrep (1)           - search compressed files for a regular expression
xzgrep (1)             - search compressed files for a regular expression
zegrep (1)              - search possibly compressed files for a regular expression
zfgrep (1)              - search possibly compressed files for a regular expression
zgrep (1)               - search possibly compressed files for a regular expression
zipgrep (1)             - search files in a ZIP archive for lines matching a pattern
```



2. Read man page of the command

- ▶ This is the **easy** and **most effective** method to get quick help on any Unix command.
- ▶ It might not have lot of practical examples, it does explain all the options available for a command in a detailed manner.
- ▶ **\$ man <Command>** (Read the man page for a Command)
 - ▶ The “man” command reads from the manual pages stored in your distribution

```

embedded_system_ks@embedded-KS:~$ man ls
embedded_system_ks@embedded-KS:~$ man ls
embedded system ks@embedded-KS:~$ man ls

○○○ embedded_system_ks@embedded-KS:~          User Commands
LS(1)                                         LS(1)

NAME
ls - list directory contents

SYNOPSIS
ls [OPTION]... [FILE]...

DESCRIPTION
List information about the FILEs (the current directory by default). Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.

-a, --all
      do not ignore entries starting with .

-A, --almost-all
      do not list implied . and ..

--author
      with -l, print the author of each file

-b, --escape
      print C-style escapes for nongraphic characters

--block-size=SIZE
      scale sizes by SIZE before printing them; e.g., '--block-size=M' prints sizes in units of
      1,048,576 bytes; see SIZE format below

-B, --ignore-backups
      do not list implied entries ending with -

-c
      with -lt: sort by, and show, ctime (time of last modification of file status information); with
      -l: show ctime and sort by name; otherwise: sort by ctime, newest first

-C
      list entries by columns

--color[=WHEN]
      colorize the output; WHEN can be 'always' (default if omitted), 'auto', or 'never'; more info
      below

-d, --directory
      list directories themselves, not their contents

Manual page ls(1) line 1 (press h for help or q to quit)

```

<https://www.learn-in-depth.com/>

<https://www.facebook.com/groups/embedded.system.KS/>

Manual sections

- The manual is generally split into eight numbered sections, organized as follows

Section Description

- | | |
|---|--|
| 1 | General commands |
| 2 | System calls |
| 3 | Library functions, covering in particular the C standard library |
| 4 | Special files (usually devices, those found in /dev) and drivers |
| 5 | File formats and conventions |
| 6 | Games and screensavers |
| 7 | Miscellanea |
| 8 | System administration commands and daemons |

```
embedded_system_ks@embedded-KS:~$ embedded_system_ks@embedded-KS:~$ man 3 malloc
embedded_system_ks@embedded-KS:~$
```

MALLOC(3) Linux Programmer's Manual **MALLOC(3)**

NAME malloc, free, calloc, realloc - allocate and free dynamic memory

SYNOPSIS

```
#include <stdlib.h>

void *malloc(size_t size);
void free(void *ptr);
void *calloc(size_t nmemb, size_t size);
void *realloc(void *ptr, size_t size);
```

DESCRIPTION

The **malloc()** function allocates **size** bytes and returns a pointer to the allocated memory. The memory is **not initialized**. If **size** is 0, then **malloc()** returns either **NULL**, or a unique pointer value that can later be successfully passed to **free()**.

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



Manual sections examples

```
embedded_system_ks@embedded-KS:~$  
embedded_system_ks@embedded-KS:~$ man 2 fork
```

```
embedded_system_ks@embedded-KS: ~          Linux Programmer's Manual          FORK(2)
FORK(2)                                     NAME
NAME                                         fork - create a child process
SYNOPSIS
#include <unistd.h>
pid_t fork(void);
DESCRIPTION
fork() creates a new process by duplicating the calling process. The new process is referred to as the child process. The calling process is referred to as the parent process.

The child process and the parent process run in separate memory spaces. At the time of fork() both memory spaces have the same content. Memory writes, file mappings (mmap(2)), and unmappings (munmap(2)) performed by one of the processes do not affect the other.

The child process is an exact duplicate of the parent process except for the following points:
```

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN IN DEPTH
#Be professional in
embedded system

45

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

Manual sections examples

```
embedded_system_ks@embedded-KS:~$  
embedded_system_ks@embedded-KS:~$ man 4 tty
```

embedded Linux

```
embedded_system_ks@embedded-KS:~  
  
TTY(4)          Linux Programmer's Manual          TTY(4)  
  
NAME  
    tty - controlling terminal  
  
DESCRIPTION  
    The file /dev/tty is a character file with major number 5 and minor number 0, usually of mode 0666 and owner.group root.tty. It is a synonym for the controlling terminal of a process, if any.  
  
    In addition to the ioctl(2) requests supported by the device that tty refers to, the ioctl(2) request TIOCNNTY is supported.  
  
TIOCNNTY  
    Detach the calling process from its controlling terminal.  
  
    If the process is the session leader, then SIGHUP and SIGCONT signals are sent to the foreground process group and all processes in the current session lose their controlling tty.  
  
    This ioctl(2) call works only on file descriptors connected to /dev/tty. It is used by daemon processes when they are invoked by a user at a terminal. The process attempts to open /dev/tty. If the open succeeds, it detaches itself from the terminal by using TIOCNNTY, while if the open fails, it is obviously not attached to a terminal and does not need to detach itself.
```

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN IN DEPTH
#Be professional in
embedded system

46

Manual sections examples

```
embedded_system_ks@embedded-KS:~$  
embedded_system_ks@embedded-KS:~$  
embedded_system_ks@embedded-KS:~$ man 3 socket  
No manual entry for socket in section 3  
embedded_system_ks@embedded-KS:~$ man 2 socket  
embedded_system_ks@embedded-KS:~$  
embedded_system_ks@embedded-KS:~$
```

Embedded Linux

ENG.KEROLES SHENOUDA

Socket is linux system call so it should be
in section 2

```
embedded_system_ks@embedded-KS:~$ SOCKET(2)          Linux Programmer's Manual           SOCKET(2)  
  
NAME  
    socket - create an endpoint for communication  
  
SYNOPSIS  
    #include <sys/types.h>      /* See NOTES */  
    #include <sys/socket.h>  
  
    int socket(int domain, int type, int protocol);  
  
DESCRIPTION  
    socket() creates an endpoint for communication and returns a file descriptor that refers to that endpoint.  
  
    The domain argument specifies a communication domain; this selects the protocol family which will be used for communication. These families are defined in <sys/socket.h>. The currently understood formats include:  
  


| Name              | Purpose                        | Man page   |
|-------------------|--------------------------------|------------|
| AF_UNIX, AF_LOCAL | Local communication            | unix(7)    |
| AF_INET           | IPv4 Internet protocols        | ip(7)      |
| AF_INET6          | IPv6 Internet protocols        | ipv6(7)    |
| AF_IPX            | IPX - Novell protocols         |            |
| AF_NETLINK        | Kernel user interface device   | netlink(7) |
| AF_X25            | ITU-T X.25 / ISO-8208 protocol | x25(7)     |
| AF_AX25           | Amateur radio AX.25 protocol   |            |
| AF_ATMPVC         | Access to raw ATM PVCs         |            |
| AF_APPLETALK      | AppleTalk                      | ddp(7)     |
| AF_PACKET         | Low level packet interface     | packet(7)  |
| AF_ALG            | Interface to kernel crypto API |            |

  
The socket has the indicated type, which specifies the communication semantics. Currently defined types are:  
  


| Type           | Description                                                                                                                                                                                         |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SOCK_STREAM    | Provides sequenced, reliable, two-way, connection-based byte streams. An out-of-band data transmission mechanism may be supported.                                                                  |
| SOCK_DGRAM     | Supports datagrams (connectionless, unreliable messages of a fixed maximum length).                                                                                                                 |
| SOCK_SEQPACKET | Provides a sequenced, reliable, two-way connection-based data transmission path for datagrams of fixed maximum length; a consumer is required to read an entire packet with each input system call. |


```

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



\$man -k <description>

- ▶ If you don't know the exact command name for a specific functionality, use **man -k** option to locate the command.
- ▶ The below example lists all the available commands that has the word "print" in the description.

```
embedded_system_ks@embedded-KS: ~ man -k print
BN_print (3ssl)      - format conversions
arch (1)             - print machine hardware name (same as uname -m)
arm-linux-gnueabi-strings (1) - print the strings of printable characters in files.
arm-none-eabi-strings (1) - print the strings of printable characters in files.
ASN1 STRING print (3ssl) - ASN1 STRING output routines.
ASN1 STRING print_ex (3ssl) - ASN1 STRING output routines.
ASN1 STRING print_ex_fp (3ssl) - ASN1 STRING output routines.
ASN1 TIME print (3ssl) - ASN.1 Time functions.
asprintf (3)          - print to allocated string
base32 (1)            - base32 encode/decode data and print to standard output
base64 (1)            - base64 encode/decode data and print to standard output
blkid (8)             - locate/print block device attributes
bn_print (3ssl)        - BIGNUM library internal functions
BN_print_fp (3ssl)    - format conversions
bsd-from (1)          - print names of those who have sent mail
caca_conio_cprintf (3caca) - The libcaca public header.
caca_conio_printf (3caca) - The libcaca public header.
caca_printf (3caca)   - libcaca canvas drawing
caca_vprintf (3caca)  - libcaca canvas drawing
cat (1)               - concatenate files and print on the standard output
cups (1)              - a standards-based, open source printing system
cups-browsed (8)       - A daemon for browsing the Bonjour broadcasts of shared, remote CUPS printers
cups-calibrate (8)     - ESP CUPS Printer Calibration Tool
cups-genppdupdate (8) - update CUPS-Gutenprint PPD files
cups-lpd (8)           - receive print jobs and report printer status to lpd clients
cupsaddsmb (8)         - export printers to samba for windows clients
cupsdisable (8)        - stop/start printers and classes
cupsenable (8)         - stop/start printers and classes
date (1)              - print or set the system date and time
dbus-monitor (1)       - debug probe to print message bus messages
DHparams print (3ssl)  - print cryptographic parameters
DHparams print_fp (3ssl) - print cryptographic parameters
dmesg (1)              - print or control the kernel ring buffer
dprintf (3)             - formatted output conversion
driverless (1)         - PPD generator utility for driverless printing
DSA print (3ssl)        - print cryptographic parameters
DSA print_fp (3ssl)    - print cryptographic parameters
DSAprams print (3ssl)  - print cryptographic parameters
DSAprams print_fp (3ssl) - print cryptographic parameters
ECPKParameters print (3ssl) - Functions for decoding and encoding ASN1 representations of elliptic curve parameters
ECPKParameters print_fp (3ssl) - Functions for decoding and encoding ASN1 representations of elliptic curve parameters
egrep (1)               - print lines matching a pattern
ERR_print_errors (3ssl) - print error messages
ERR_print_errors_fp (3ssl) - print error messages
error_print_programme (3) - glibc error reporting functions
evince-previewer (1)    - show a printing preview of PostScript and PDF documents
EVP_PKEY print_params (3ssl) - public key algorithm printing routines.
EVP_PKEY print_private (3ssl) - public key algorithm printing routines.
EVP_PKEY print_public (3ssl) - public key algorithm printing routines.
FcFontSetPrint (3)      - Print a set of patterns to stdout
FcPatternPrint (3)      - Print a pattern for debugging
FcValuePrint (3)         - Print a value to stdout
fdtdump (1)             - prints a readable version of a flat device-tree file.
fgconsole (1)            - print the number of the active VT.
fgrep (1)               - print lines matching a pattern
fprintf (3)              - formatted output conversion
fprintf_3avr (3avr)     - <stdio.h>: Standard I/O facilities
fprintf_P (3avr)         - <stdio.h>: Standard I/O facilities
from (1)                - print names of those who have sent mail
fuprintf (3)             - formatted wide-character output conversion
getkeycodes (8)          - print kernel scancode-to-keycode mapping table
git-grep (1)             - Print lines matching a pattern
grep (1)                 - print lines matching a pattern
groups (1)               - print the groups a user is in
gsbj (1)                 - Format and print text for BubbleJet printer using ghostscript
gsdj (1)                 - Format and print text for DeskJet printer using ghostscript
gsdj500 (1)              - Format and print text for DeskJet 500 BubbleJet using ghostscript
gslj (1)                 - Format and print text for LaserJet printer using ghostscript
gslp (1)                 - Format and print text using ghostscript
gst-inspect-1.0 (1)       - print info about a GStreamer plugin or element
gst-typefind-1.0 (1)     - print Media type of file
gtk-query-settings (1)   - Utility to print name and value of all GtkSettings properties
hostid (1)                - print the numeric identifier for the current host
hp-align (1)              - Printer Cartridge Alignment Utility
hp-clean (1)              - Printer Printhead Cleaning Utility
hp-colorcal (1)           - Printer Cartridge Color Calibration Utility
hp-config_usb_printer (1) - HP device config using USB
hp-probe (1)              - Printer Discovery Utility
hp-setup (1)              - Printer/Fax Setup Utility
hp-testpage (1)            - Testpage Print Utility
i686-linux-gnu-strings (1) - print the strings of printable characters in files.
id (1)                   - print real and effective user and group IDs
infocmp (1)              - compare or print out terminfo descriptions
ippfind (1)               - find internet printing protocol printers
ippserver (1)             - a simple internet printing protocol server
ippool (1)                - perform internet printing protocol requests
ippusbxd (1)              - userland driver for IPP-over-USB class printers
isprint (3)               - character classification functions
isprint_3avr (3avr)        - <cctype.h>: Character Operations
isprint_l (3)              - character classification functions
iswprint (3)               - test for printing wide character
jmap (1)                  - Prints shared object memory maps or heap memory details for a process, core file, or remote debug server...
jstack (1)                - Prints Java thread stack traces for a Java process, core file, or remote debug server...
ldd (1)                   - print shared object dependencies
logname (1)               - print user's login name
lp (1)                     - print files
lp (4)                     - line printer devices
lpadmin (8)                - configure cups printers and classes
lpc (8)                    - line printer control program
lpoptions (1)              - display or set printer options and defaults
lpq (1)                    - show printer queue status
lprm (1)                   - print files
lpsstat (1)                - cancel print jobs
lsof (1)                   - print cups status information
lsb_release (1)             - print distribution-specific information
malloc_stats (3)           - print memory allocation statistics
minfo (1)                  - print the parameters of a MSDOS filesystem
Net::DBus::Dumper (3pm)    - Stringify Net::DBus objects suitable for printing
netstat (8)                - Print network connections, routing tables, interface statistics, masquerade connection...
nfld_sf (8)                - load and unload os fingerprint database
nproc (1)                  - print the number of processing units available
ntfscat (8)                - print NTFS files and streams on the standard output
oakdecode (1)               - Decode an OAKT printer stream into human readable form.
```

<https://www.learn-in-depth.com/>

<https://www.facebook.com/groups/embedded.system.KS/>

3. Display Single Line Description About a Unix Command

- ▶ To display a single line description about the command, you can use whatis command.
- ▶ \$ whatis UNIXCOMMAND

```
embedded_system_ks@embedded-KS:~$ whatis ls
ls (1)                      - list directory contents
embedded_system_ks@embedded-KS:~$ whatis fork
fork (2)                     - create a child process
embedded_system_ks@embedded-KS:~$ whatis pthread_create
pthread_create (3)           - create a new thread
embedded_system_ks@embedded-KS:~$ whatis socket
socket (2)                   - create an endpoint for communication
socket (7)                   - Linux socket interface
embedded_system_ks@embedded-KS:~$ whatis write
write (2)                    - write to a file descriptor
write (1)                    - send a message to another user
embedded_system_ks@embedded-KS:~$ whatis man
man (7)                      - macros to format man pages
man (1)                      - an interface to the on-line reference manuals
embedded_system_ks@embedded-KS:~$
```

Embedded Linux

ENG. KEROLES SHENOUDA

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



4. Use -h or --help option of the command itself

- ▶ Sometimes you may know the functionality of a command very well, but cannot recollect all the available options for a specific command.
- ▶ Use **--help** option of the command to review all available options of the command.

```
embedded_system_ks@embedded-KS:~$ ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.
-a, --all                  do not ignore entries starting with .
-A, --almost-all           do not list implied . and ..
--author                   with -l, print the author of each file
-b, --escape                print C-style escapes for nongraphic characters
--block-size=SIZE           scale sizes by SIZE before printing them; e.g.,
                           '--block-size=M' prints sizes in units of
                           1,048,576 bytes; see SIZE format below
-B, --ignore-backups       do not list implied entries ending with ~
-c                         with -lt: sort by, and show, ctime (time of last
                           modification of file status information);
                           with -l: show ctime and sort by name;
                           otherwise: sort by ctime, newest first
-C                         list entries by columns
                           colorize the output; WHEN can be 'always' (default
                           if omitted), 'auto', or 'never'; more info below
-d, --directory            list directories themselves, not their contents
-D, --dired                 generate output designed for Emacs' dired mode
-f                         do not sort, enable -aU, disable -ls --color
-F, --classify              append indicator (one of */=>@]) to entries
                           likewise, except do not append '*'
                           across -x, commas -m, horizontal -x, long -l,
                           single-column -l, verbose -l, vertical -C
                           like -l --time-style=full-iso
-g                         like -l, but do not list owner
--group-directories-first  group directories before files;
                           can be augmented with a --sort option, but any
                           use of --sort=none (-U) disables grouping
-G, --no-group              in a long listing, don't print group names
-h, --human-readable        with -l and/or -s, print human readable sizes
```



5. Read Info Documents using Unix info Command

- ▶ When you are not able to find the required information from the Unix man page, try the info documents using the Unix info command as shown below.
- ▶ \$info <command>

```
embedded_system_ks@embedded-KS:~$ info ls
embedded_system_ks@embedded-KS:~$ info ls
embedded_system_ks@embedded-KS:~$
```

Next: dir invocation, Up: Directory listing

10.1 'ls': List directory contents

The 'ls' program lists information about files (of any type, including directories). Options and file arguments can be intermixed arbitrarily, as usual.

For non-option command-line arguments that are directories, by default 'ls' lists the contents of directories, not recursively, and omitting files with names beginning with '.'. For other non-option arguments, by default 'ls' lists just the file name. If no non-option argument is specified, 'ls' operates on the current directory, acting as if it had been invoked with a single argument of '.'.

By default, the output is sorted alphabetically, according to the locale settings in effect.(1) If standard output is a terminal, the output is in columns (sorted vertically) and control characters are output as question marks; otherwise, the output is listed one per line and control characters are output as-is.

Because 'ls' is such a fundamental program, it has accumulated many options over the years. They are described in the subsections below; within each section, options are listed alphabetically (ignoring case). The division of options into the subsections is not absolute, since some options affect more than one aspect of 'ls''s operation.

Exit status:

- 0 success
- 1 minor problems (e.g., failure to access a file or directory not specified as a command line argument. This happens when listing a directory in which entries are actively being removed or renamed.)
- 2 serious trouble (e.g., memory exhausted, invalid option, failure to access a file or directory specified as a command line argument or a directory loop)



[#LEARN IN DEPTH](#)

[#Be professional in
embedded system](#)

51

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

Composite Commands

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



<https://www.facebook.com/groups/embedded.system.KS/>

eng. Keroles Shenouda

#LEARN IN DEPTH
#Be professional in
embedded system

52

Composite Commands

- ▶ build Composite Commands through the following:
 - ▶ Sequential Commands
 - ▶ Conditional Commands
 - ▶ Command Loops
 - ▶ Input/Output Redirection
 - ▶ Pipes
 - ▶ Command Argument Expansion
 - ▶ Command Argument Quoting

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN IN DEPTH

#Be professional in
embedded system

53

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

SEQUENTIAL COMMANDS

A; B – Run A and then B, regardless of the success or failure of A

- ▶ Combining two or more commands on the command line is also known as **"command chaining"**
- ▶ **The Semicolon (:) Operator**
- ▶ The semicolon (:) operator allows you to execute multiple commands in succession, regardless of whether each previous command succeeds.

```
embedded_system_ks@embedded-KS:~  
embedded_system_ks@embedded-KS:~$ ls ; pwd ; whoami  
a.out      Documents      labs      Music      Public      temp      two_process_alarm.c  
Captsdure.PNG  Downloads      log2.text  Pictures    qemu       Templates  Videos  
Desktop      examples.desktop  log.text   pthread_lab.c share     trampoline workspace  
/home/embedded_system_ks  
embedded_system_ks  
embedded_system_ks@embedded-KS:~$
```

Embedded Linux

ENG. KEROLES SHENOUDA

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

CONDITIONAL COMMANDS

- ▶ The Logical AND Operator (`&&`)
 - ▶ If you want the second command to **only run** if the first command is successful, separate the commands with the logical AND operator

```
embedded_system_ks@embedded-KS:~/MyFolder
embedded_system_ks@embedded-KS:~$ mkdir MyFolder && cd MyFolder
embedded_system_ks@embedded-KS:~/MyFolder$
```

Embedded Linux
ENG. KEROLES SHENOUDA

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN IN DEPTH

#Be professional in
embedded system

55

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

CONDITIONAL COMMANDS

A || B – Run B only if A failed

- Sometimes you might want to execute a second command only if the first command does not succeed.
- To do this, we use the logical OR operator

```
embedded_system_ks@embedded-KS: ~  
embedded_system_ks@embedded-KS: ~$ [ -d ~/MyFolder ] || mkdir ~/MyFolder  
embedded_system_ks@embedded-KS: ~$
```

check to see if the MyFolder directory exists

and create it if it doesn't



#LEARN IN DEPTH

#Be professional in
embedded system

56

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

Combining Multiple Operators

- ▶ You can combine multiple operators on the command line, too.

```
embedded_system_ks@embedded-KS: ~
embedded_system_ks@embedded-KS:~$ [ -f ~/sample.txt ] && echo "File exists." || touch ~/sample.txt
embedded_system_ks@embedded-KS:~$ [ -f ~/sample.txt ] && echo "File exists." || touch ~/sample.txt
"File exists."
embedded_system_ks@embedded-KS:~$
```

Embedded Linux

ENG. KEROLES SHENOUDA

check if a file exists ([-f ~/sample.txt]). If it does, we print a message to the screen saying so (echo "File exists."). If not, we create the file (touch ~/sample.txt)

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



Command Loops

- ▶ A 'for loop' is a bash programming language statement which allows code to be repeatedly executed. A for loop is classified as an iteration statement

```
for VARIABLE in 1 2 3 4 5 .. N
do
    command1
    command2
    commandN
done
```

```
#!/bin/bash
echo "Bash version ${BASH_VERSION}..."
for i in {0..10..2}
do
    echo "Welcome $i times"
done
```

```
Bash version 4.0.33(0)-release...
Welcome 0 times
Welcome 2 times
Welcome 4 times
Welcome 6 times
Welcome 8 times
Welcome 10 times
```

```
#!/bin/bash
for (( ; ; ))
do
    echo "infinite loops [ hit CTRL+C to stop]"
done
```

```
for VARIABLE in file1 file2 file3
do
    command1 on $VARIABLE
    command2
    commandN
done
```

```
#!/bin/bash
for file in /etc/*
do
    if [ "${file}" == "/etc/resolv.conf" ]
    then
        countNameservers=$(grep -c nameserver /etc/resolv.conf)
        echo "Total ${countNameservers} nameservers defined in ${file}"
        break
    fi
done
```



<https://www.facebook.com/groups/embedded.system.KS/>

eng. Keroles Shenouda

#LEARN IN DEPTH
#Be professional in
embedded system

58

I/O Redirection with files

- ▶ Standard Output Redirection
 - ▶ \$<command> > file (Redirect Output to file; Overwrite)
 - ▶ \$<command> >> file (Redirect Output to file; Append)
- ▶ Standard Error Redirection
 - ▶ \$<command> 2> file (Redirect error to file; Overwrite)
 - ▶ \$<command> 2>> file (Redirect error to file; Append)
- ▶ Both Output & Error Redirection
 - ▶ \$<command> &> file (Redirect output/error to file; Overwrite)
 - ▶ \$<command> &>> file (Redirect output/error to file; Append)

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN IN DEPTH

#Be professional in
embedded system

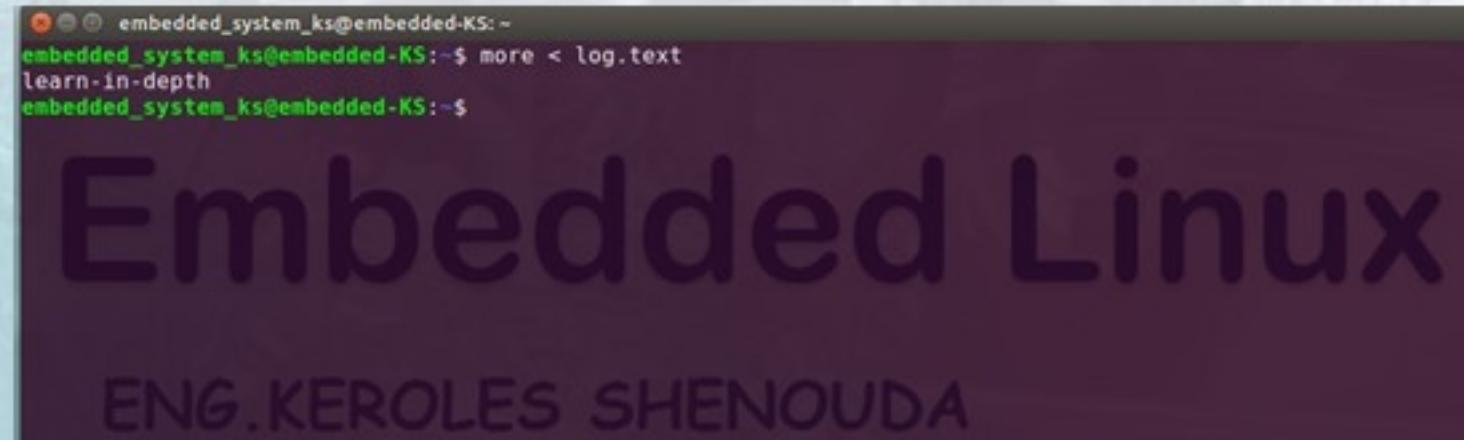
59

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

Standard Input Redirection

- ▶ \$command < file (input to the command is read from the file)
- ▶ Examples:



<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Common Devices for Redirection

- ▶ **/dev/stdout** : Standard **output** device
- ▶ **/dev/stderr**: Standard **error** device
- ▶ **/dev/stdin**: Standard **input** device
- ▶ **/dev/null** : This device is useful for being **a data sink**.
- ▶ **/dev/zero** : This device is useful as an **input device to generate and infinite stream of zeros**
- ▶ **/dev/random** : This device is useful as an **input device to generate random bytes**.
- ▶ **/dev/urandom**: This device is useful as an **input device to generate quasi-random bytes**.

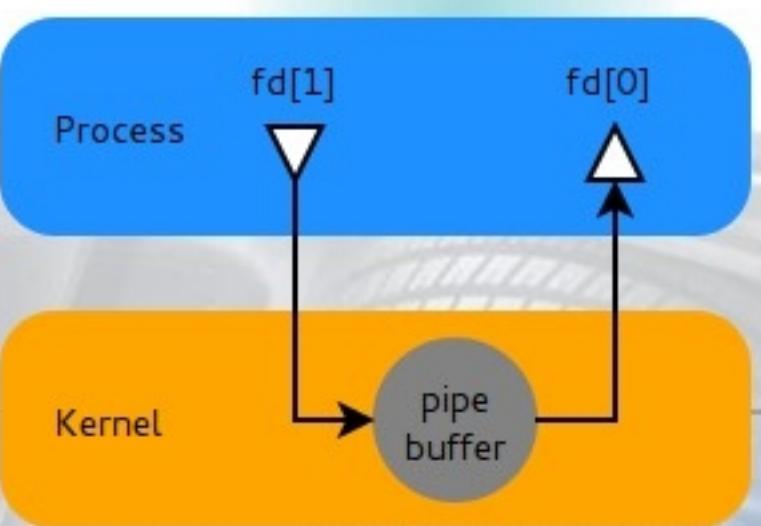
<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



Press here

USING PIPES

- ▶ The Pipe is a command in Linux that lets you use two or more commands such that output of one command serves as input to the next.
- ▶ A Pipe is a mechanism in Linux Kernel that is used to enable one process to send information to the other process
- ▶ This is called IPC (Inter Process Communication)



The contents of the 'sample' file

```
home@VirtualBox:~$ cat sample
Bat
Goat
Apple
Dog
First
Eat
Hide
```

Using 'grep' for searching Apple

```
home@VirtualBox:~$ cat sample | grep Apple
Apple
```

Using 'grep' for searching Eat

```
home@VirtualBox:~$ cat sample | grep Eat
Eat
```



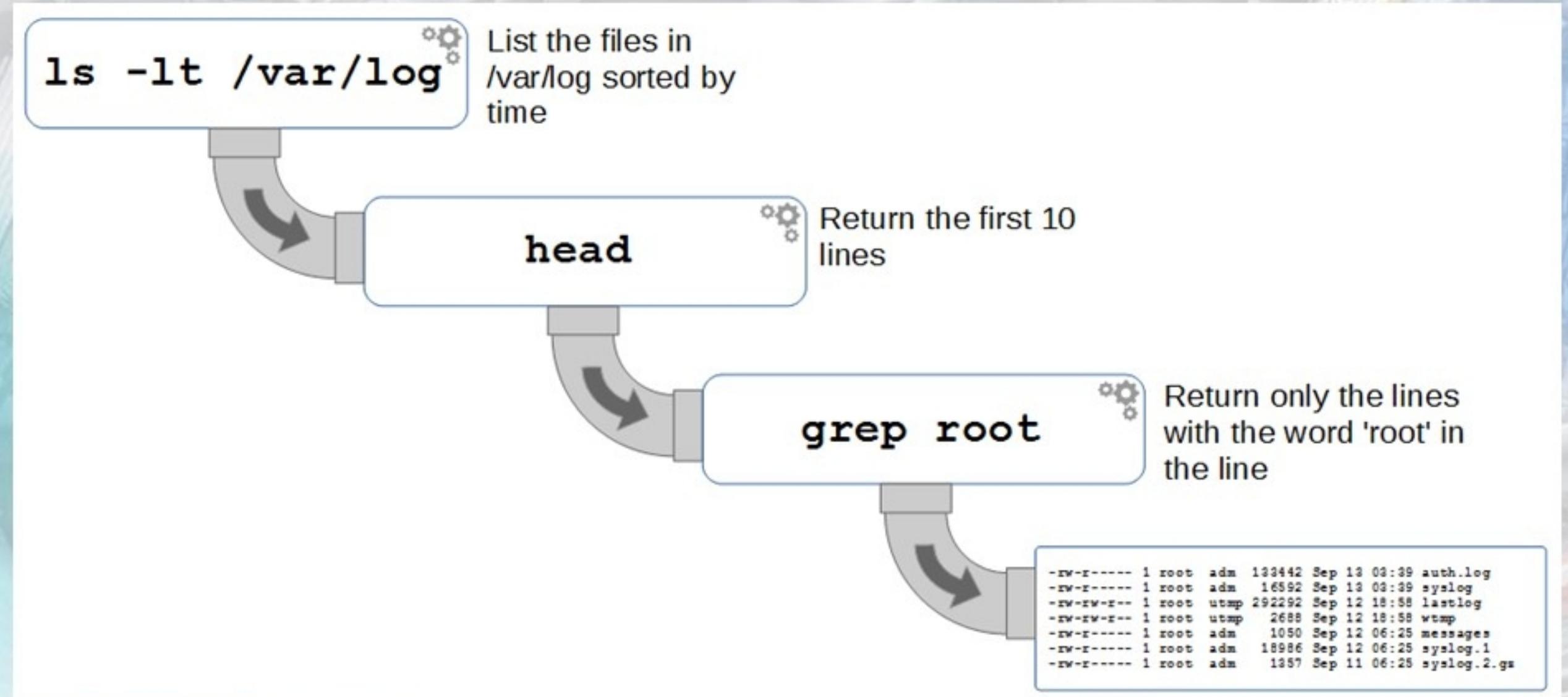
#LEARN IN DEPTH
#Be professional in
embedded system

62

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

USING PIPES example



<https://www.learn-in-depth.com/>

<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN IN DEPTH
#Be professional in
embedded system

63

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

tee Command

- ▶ The tee command sends the output to a set of files as well as the standard stdout (default is screen)
- ▶ \$ <command> | tee <sink>
- ▶ Examples
 - ▶ \$make -a |tee log.txt

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Commands Arguments Expansion Arithmetic Expansion (\$(()))

- ▶ The Arithmetic Expansion is used to put arithmetic expressions to be evaluated

```
embedded_system_ks@embedded-KS:~$ echo $((5 + 6))
11
embedded_system_ks@embedded-KS:~$ echo $(( ((5**2)) * 3 ))
75
embedded_system_ks@embedded-KS:~$
```

Embedded Linux

ENG.KEROLES SHENOUDA

Commands Arguments Quoting Back Quotes (` `)

- ▶ Back Quotes are used to do commands **inside** commands/strings
- ▶ Examples:
 - ▶ `$ cd /lib/modules/`uname -r``

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN IN DEPTH
#Be professional in
embedded system

eng. Keroles Shenouda
<https://www.facebook.com/groups/embedded.system.KS/>

<https://www.learn-in-depth.com/>

<https://www.facebook.com/groups/embedded.system.KS/>

Eng.keroles.karam@gmail.com

Shell script

Eng. Keroles Shenouda

Embedded Linux

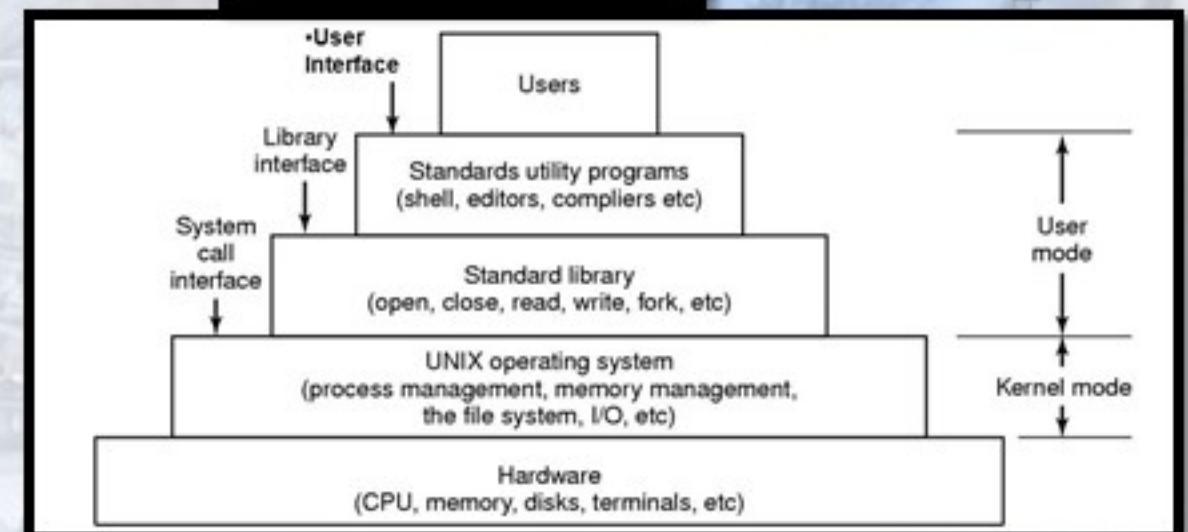
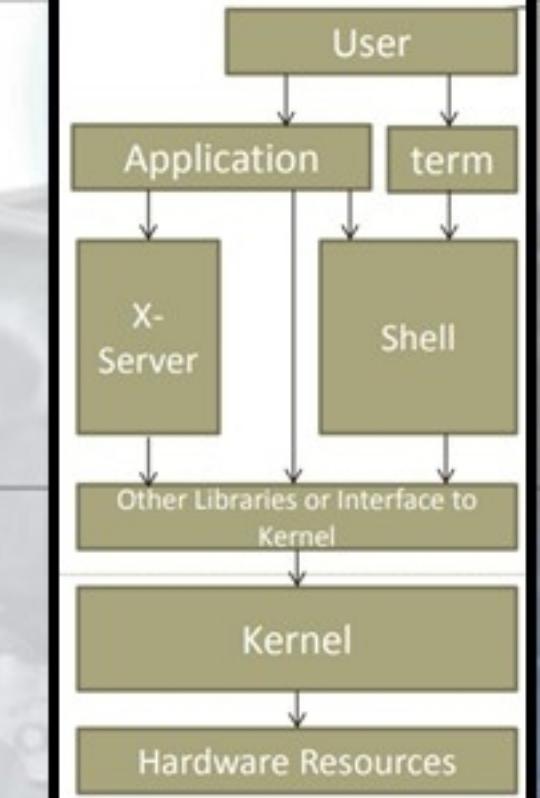
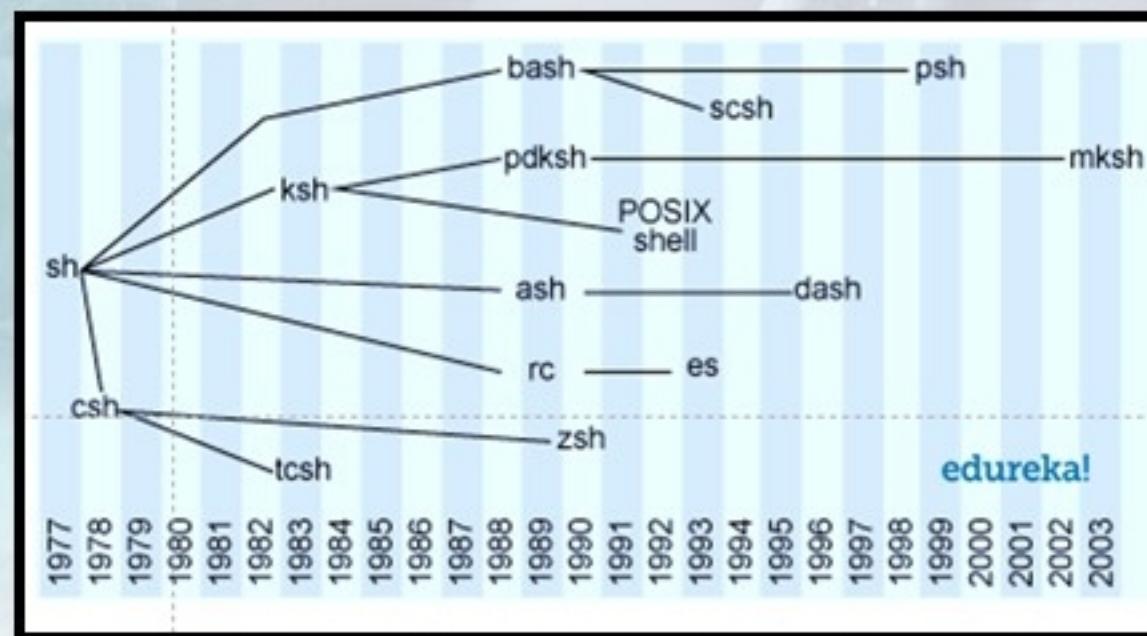
<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Eng.keroles.karam@gmail.com



What is the “Shell”

- ▶ The Shell is a user space program that accepts text input
- ▶ It is accessed through a terminal or terminal emulator program
- ▶ "a shell manages the interaction between the system and its users"
- ▶ Types of Shells



<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN IN DEPTH
#Be professional in
embedded system

eng. Keroles Shenouda
<https://www.facebook.com/groups/embedded.system.KS/>

<https://www.learn-in-depth.com/>

<https://www.facebook.com/groups/embedded.system.KS/>

Eng.keroles.karam@gmail.com

Eng. Keroles Shenouda

Embedded Linux

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Eng. Keroles Shenouda

Eng. Keroles Shenouda

Eng.keroles.karam@gmail.com

Eng. Keroles Shenouda



#LEARN IN DEPTH
#Be professional in
embedded system

69

eng. Keroles SH

<https://www>

Searching Text (grep Command)

- ▶ A powerful text search utility
- ▶ Used to find pattern in files or text
- ▶ Type man grep to find list of options
- ▶ **grep [-options] pattern [filename]**

Examples:

```
$ grep pattern filename
$ grep pattern file1 file2
$ grep -i desktop /etc/services
$ grep [yf] /etc/group
$ grep -vi tcp /etc/services
$ ip addr show | grep inet
```

Option	Description
-b	Display the block number at the beginning of each line.
-c	Display the number of matched lines.
-h	Display the matched lines, but do not display the filenames.
-i	Ignore case sensitivity.
-l	Display the filenames, but do not display the matched lines.
-n	Display the matched lines and their line numbers.
-s	Silent mode.
-v	Display all lines that do NOT match.

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



Press
here

#LEARN IN DEPTH

#Be professional in
embedded system

70

eng. Keroles Shenouda

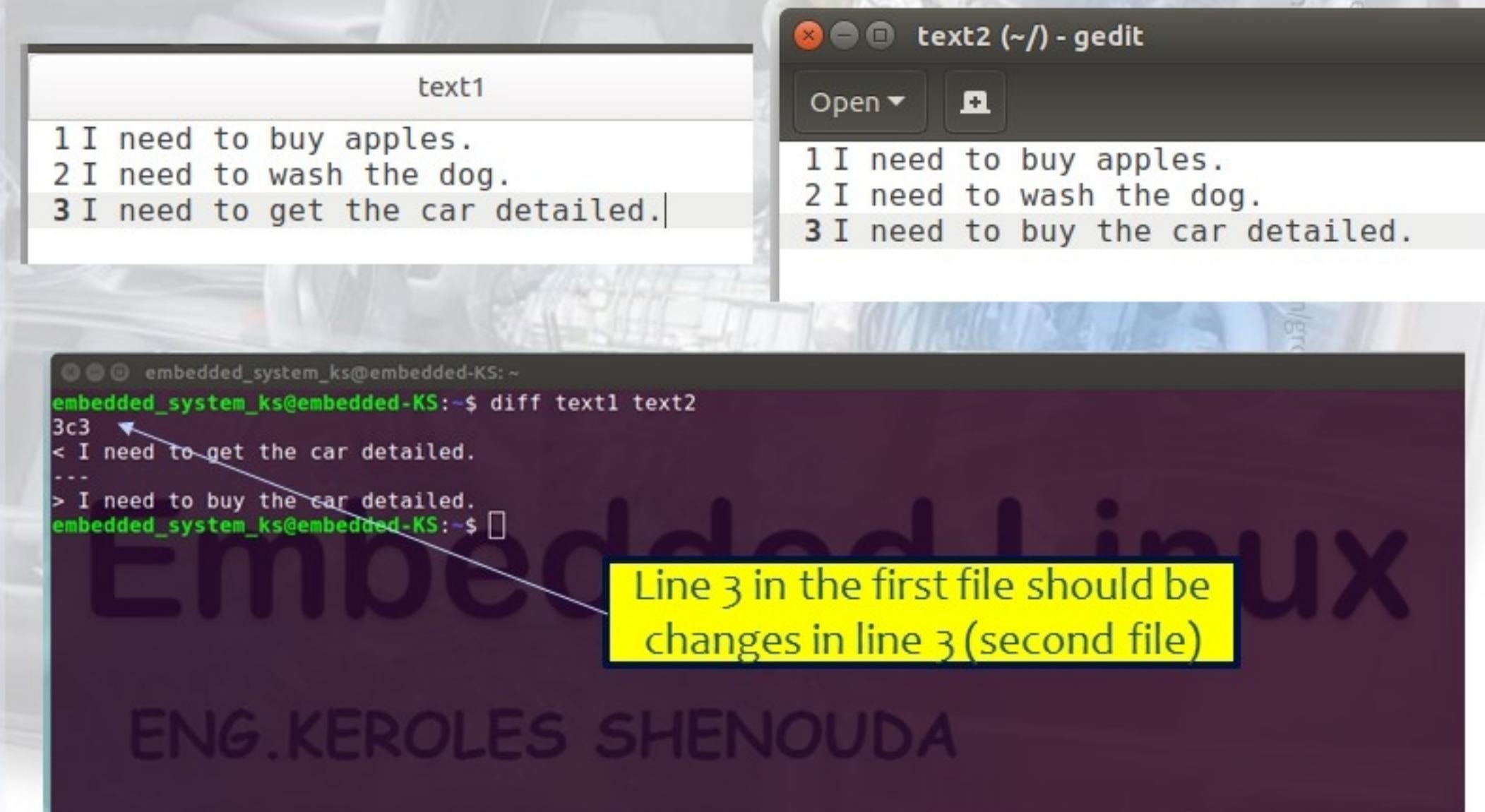
<https://www.facebook.com/groups/embedded.system.KS/>

Comparing Text Files

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

Comparing Text Files (diff Command)

- ▶ Diff is simple and easy to use, it comes pre-installed on most Linux distributions.
- ▶ It compares files line by line and outputs the difference between them.
- ▶ `$ man diff`



```

text1
1 I need to buy apples.
2 I need to wash the dog.
3 I need to get the car detailed.

text2 (~/) - gedit
Open + 1 I need to buy apples.
2 I need to wash the dog.
3 I need to buy the car detailed.

embedded_system_ks@embedded-KS:~$ diff text1 text2
3c3
< I need to get the car detailed.
...
> I need to buy the car detailed.
embedded_system_ks@embedded-KS:~$ 
```

Line 3 in the first file should be changes in line 3 (second file)

ENG.KEROLES SHENOUDA

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



FOLLOW US

Press here

#LEARN IN DEPTH

#Be professional in
embedded system

72

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

How to Read diff Output

► The First Line of Output

- The first line in the output indicates the line numbers that contain differences and the type of changes that have been made.

```
11,12c11
< this is text from the original file\
< \
---
> this is the same line from the second, changed file\
```

- In the example image above, 11,12 would indicate that there are changes on lines 11 - 12.
- The “c” that separates the line numbers in the example above indicates the type of change that occurred.

Letter	Meaning
c	Content was replaced
a	Content was added or appended
d	Content was deleted

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



<https://www.facebook.com/groups/embedded.system.KS/>

eng. Keroles Shenouda



#LEARN IN DEPTH
#Be professional in
embedded system

73

What is patch?

- ▶ patch is a command that takes the output from the diff and puts it into a file. Then, it can take the filed output and overwrite another file with with the changes.
- ▶ **patch Options**

-b	Creates a backup of the original file
-i	Forces the command to read the patch from the .patch file instead of from standard input
-R	Reverses the previous patch
-s	Runs the command silently. It will only show process if there are errors

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN IN DEPTH

#Be professional in
embedded system

74

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

Patch file

► Creating a patch

```
diff -u file1.html file2.html > patchfile.patch
```

► Overwrite files with changes

```
patch file1.html patchfile.patch
```

- Replace file1.html with your original file. This would overwrite the old contents of file1.html with the changed contents of file2.html.

► to Reverse a patch

```
patch -p0 -R -i patchfile.patch
```

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN IN DEPTH
#Be professional in
embedded system

75

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

Patch Directories

- ▶ Using diff and patch on whole directories is a similar process to using it on single files.
- ▶ The first step is to create a patch file by using the command:

```
diff -ruN folder1/ folder2/ > patchfile.patch
```

- ▶ Then, you would issue the command to patch the original folder, which in this case is folder1:

```
patch -s -p0 < patchfile.patch
```

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN IN DEPTH
#Be professional in
embedded system

eng. Keroles Shenouda
<https://www.facebook.com/groups/embedded.system.KS/>

76

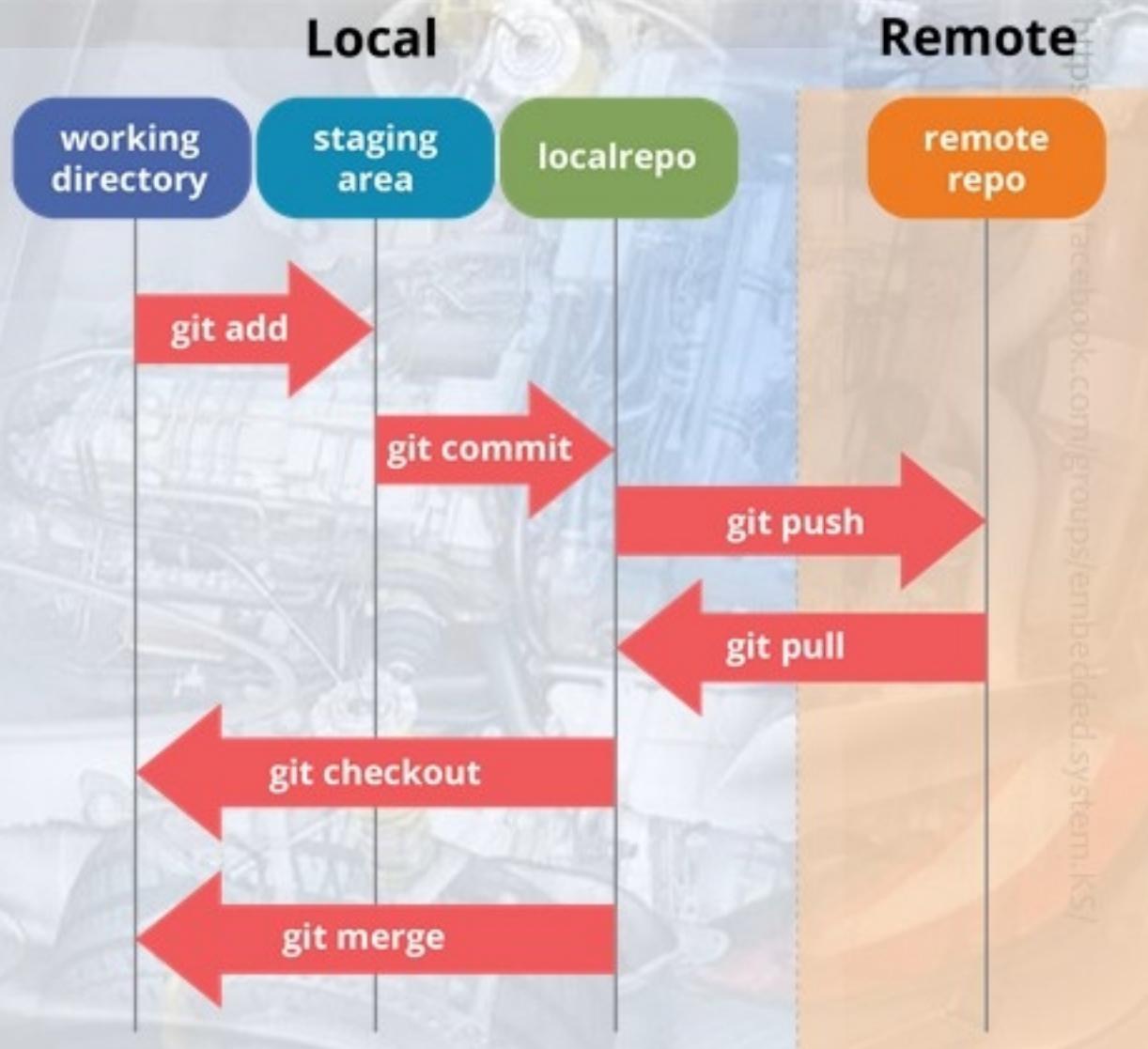
Git

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



git

- ▶ Some of the basic operations in Git are:
 - ▶ Initialize
 - ▶ Add
 - ▶ Commit
 - ▶ Pull
 - ▶ Push
- ▶ Some advanced Git operations are:
 - ▶ Branching
 - ▶ Merging



<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN IN DEPTH

#Be professional in
embedded system

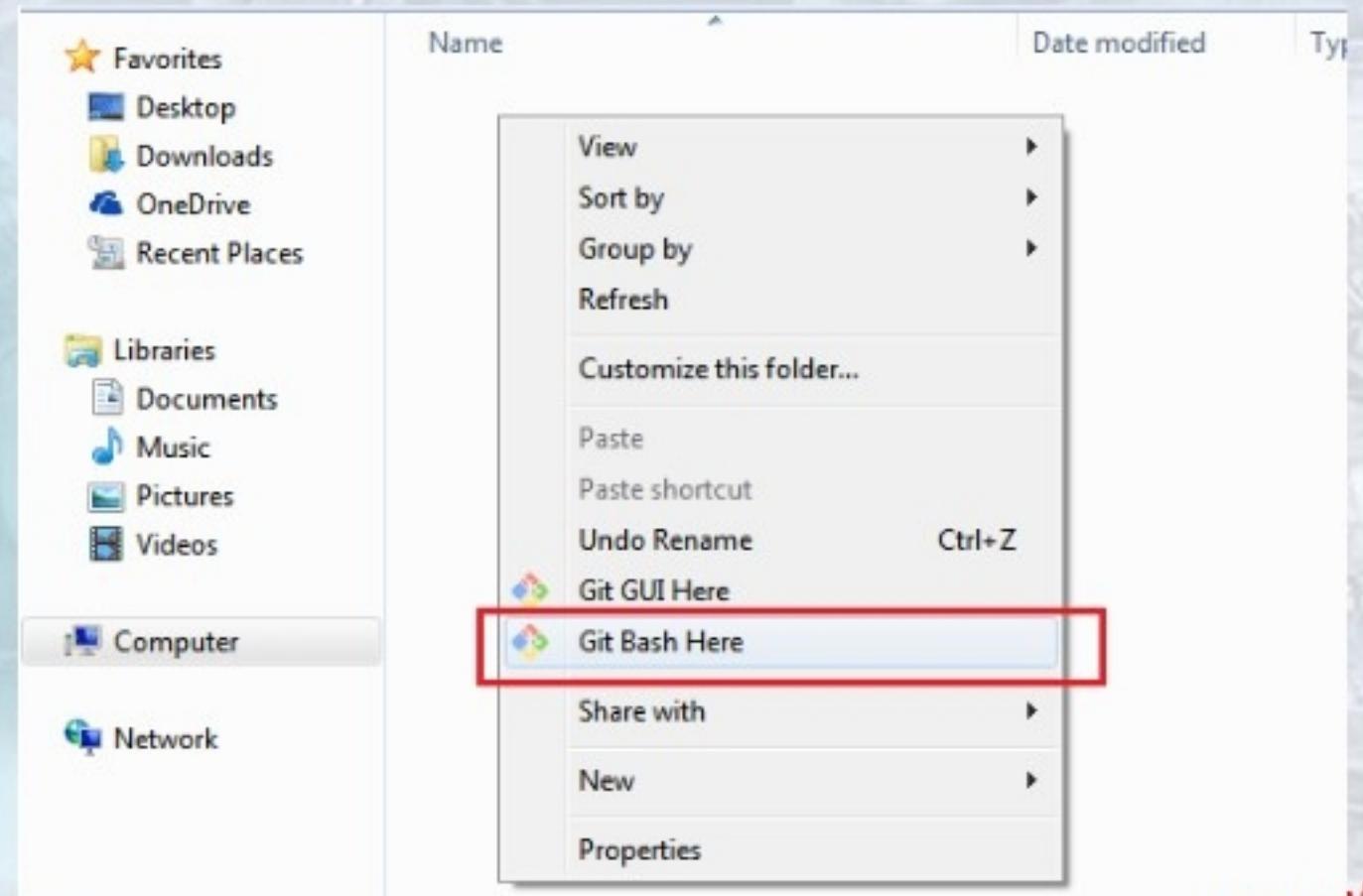
78

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

Git available in linux and windows

- ▶ After installing Git in your Windows system, just open your folder/directory where you want to store all your project files; right click and select '*Git Bash here*'



<http://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



Git Initialize

- ▶ **git init** creates an empty Git repository or re-initializes an existing one. It basically creates a **.git** directory with sub directories and template files. Running a **git init** in an existing repository will not overwrite things that are already there. It rather picks up the newly added templates.

```
embedded_system_ks@embedded-KS:~$ mkdir new_repo
embedded_system_ks@embedded-KS:~$ cd new_repo/
embedded_system_ks@embedded-KS:~/new_repo$ git init
Initialized empty Git repository in /home/embedded_system_ks/new_repo/.git/
embedded_system_ks@embedded-KS:~/new_repo$
```

```
embedded_system_ks@embedded-KS:~/embedded_linux_1$ git config --global user.email "eng.keroles.karam@gmail.com"
embedded_system_ks@embedded-KS:~/embedded_linux_1$ git config --global user.name "keroles"
embedded_system_ks@embedded-KS:~/embedded_linux_1$ git commit -m "the first version from file1.tut" file1.tut
```

<https://www.learn-in-depth.com/>

<https://www.facebook.com/groups/embedded.system.KS/>



Press here

#LEARN IN DEPTH
#Be_professional_in_embedded_system

<https://www.facebook.com/groups/embedded.system.KS/>

Git clone

keroles / embedded_linux_1

No description, website, or topics provided.

Manage topics

2 commits 1 branch 0 releases 1 contributor

Branch: master New pull request

Set up Actions Dismiss

Clone with HTTPS Use SSH

Use Git or checkout with SVN using the web URL.

HTTP://github.com/keroles/embedded_linux_1.git

Open in Desktop Download ZIP

embedded_linux_1

```
embedded_system_ks@embedded-KS:~$ git clone https://github.com/keroles/embedded_linux_1.git
Cloning into 'embedded_linux_1'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
embedded_system_ks@embedded-KS:~$ cd embedded_linux_1/
embedded_system_ks@embedded-KS:~/embedded_linux_1$ ls
README.md
```

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



81

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>#LEARN IN DEPTH
#Be professional in
embedded system

Git status

- ▶ The **git status** command lists all the modified files which are ready to be added to the local repository.
- ▶ This shows that I have one file which is not added to the index yet. This means I cannot commit changes with this file unless I have added them explicitly in the index.

```
embedded_system_ks@embedded-KS:~/embedded_linux_1$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    file1.txt
nothing added to commit but untracked files present (use "git add" to track)
```

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN IN DEPTH
#Be professional in
embedded system

82

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

Git add

- ▶ This command updates the index using the current content found in the working tree and then prepares the content in the staging area for the next commit.
- ▶ Now that the new file is added to the index, you are ready to commit it.

```
embedded_system_ks@embedded-KS:~/embedded_linux_1$ git add file1.txt
```

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN IN DEPTH
#Be professional in
embedded system

83

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

Git Commit

- ▶ This will commit the staged snapshot and will launch a text editor prompting you for a commit message.
- ▶ Or you can use:
 - ▶ `git commit -m "<message>"`

```
embedded_system_ks@embedded-KS:~/embedded_linux_1$ git commit -m "the first version from file1.txt"
[master 23bfb8a] the first version from file1.txt
 1 file changed, 1 insertion(+)
 create mode 100644 file1.txt
```

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN IN DEPTH
#Be_professional_in_embedded_system

84

eng. Keroles Shenouda

<https://www.facebook.com>

git push

- ▶ This pushes the changes from the local repository to the remote repository along with all the necessary commits and internal objects.

```
embedded_system_ks@embedded-KS:~/embedded_linux_1$ git push
Username for 'https://github.com': keroles
Password for 'https://keroles@github.com':
Counting objects: 3, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 306 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/keroles/embedded_linux_1.git
  258700c..23bfb8a  master -> master
```

The screenshot shows a GitHub repository page for 'keroles / embedded_linux_1'. The repository has 2 commits, 1 branch (master), and 1 contributor (keroles). The commit history includes:

- keroles the first version from file1.txt (12 minutes ago)
- Initial commit (12 minutes ago)
- the first version from file1.txt (8 minutes ago)
- README.md (Latest commit: 23bfb8a 8 minutes ago)

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN IN DEPTH
#Be professional in
embedded system

85

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

gitk

```
embedded_system_ks@embedded-KS:~/embedded_linux_1$ gitk
```

File Edit View Help

master remotes/origin/master the first version from file1.txt
Initial commit

keroles <eng.keroles.karam@gmail.co> 2019-12-28 03:03:33
keroles <eng.keroles.karam@gmail.co> 2019-12-28 02:59:20

SHA1 ID: 23bfb8ae264044bcb2c9bbdc7956ff6872485400 ← → Row 1/ 2

Find ↓ ↑ commit containing: Search

Diff Old version New version Lines of context: 3 Ignore space change

Author: keroles <eng.keroles.karam@gmail.com> 2019-12-28 03:03:33
Committer: keroles <eng.keroles.karam@gmail.com> 2019-12-28 03:03:33
Parent: 258700c1850ca04e43b6f9a9285ad7843272eae3 (Initial commit)
Branches: master, remotes/origin/master
Follows:
Precedes:

the first version from file1.txt

----- file1.txt -----
new file mode 100644
index 000000..45806e8
@@ -0,0 +1 @@
+learn-in-depth

Patch Tree

Comments
file1.txt

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



#LEARN IN DEPTH
#Be professional in
embedded system

86

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>

gitlog

- ▶ This shows the history of the project with a full list of the commits

```
embedded_system_ks@embedded-KS:~/embedded_linux_1$ git log
commit 23bfb8ae264044bcb2c9bbdc7956ff6872485400
Author: keroles <eng.keroles.karam@gmail.com>
Date:   Sat Dec 28 03:03:33 2019 +0200

    the first version from file1.txt

commit 258700c1850ca04e43b6f9a9285ad7843272eae3
Author: keroles <eng.keroles.karam@gmail.com>
Date:   Sat Dec 28 02:59:20 2019 +0200

    Initial commit
embedded_system_ks@embedded-KS:~/embedded_linux_1$
```

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

THANK
YOU!

Learn-in-depth.com

References

- ▶ Embedded Linux training
 - ▶ <https://bootlin.com/training/>
- ▶ [Linux kernel and driver development training](#)
- ▶ [Yocto Project and OpenEmbedded development training](#)
- ▶ [Buildroot development training](#)
- ▶ Building Embedded Linux Systems, 2nd Edition
- ▶ Mastering Embedded Linux Programming - Second Edition
- ▶ Christopher Hallinan, **Embedded Linux Primer**, Prentice Hall, 2006.
- ▶ Silberschatz, Galvin, and Gagne's **Operating System Concepts**, Eighth Edition.
- ▶ Robert love, linux kernel development 3rd edition 2010
- ▶ Advanced Linux Programming By Mark Mitchell, Jeffrey Oldham, Alex Samuel
- ▶ [Linux OS in Embedded Systems & Linux Kernel Internals\(2/2\)](#)
 - ▶ Memory Management, Paging, Virtual Memory, File system and its implementation, Secondary Storage(HDD), I/O systems
- ▶ [Ahmed ElArabawy Courses at http://linux4embeddedsystems.com/](#)

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>

- ▶ [Memory Management article](#)
- ▶ [Introduction to Operating Systems Class 9 - Memory Management](#)
- ▶ [Virtual Memory lecture for Introduction to Computer Architecture at Uppsala University.](#)
- ▶ [OS Lecture Note 13 - Address translation, Caches and TLBs](#)
- ▶ [CSE 331 Operating Systems Design lectures](#)
- ▶ [CSE 331 Virtual Memory](#)
- ▶ [CSE 332 Computer Organization and Architecture](#)
- ▶ [File Systems: Fundamentals.](#)
- ▶ [Linux System Programming](#)
- ▶ [System Calls, POSIX I/O](#)
CSE 333 Spring 2019, Justin Hsia
- ▶ [Linux basic commands](#)

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>



90

eng. Keroles Shenouda

<https://www.facebook.com/groups/embedded.system.KS/>#LEARN IN DEPTH
#Be_professional_in_embedded_system

- ▶ [File Permissions in Linux/Unix with Example](#)
- ▶ [Introduction to UNIX / Linux - 4](#)
- ▶ [12.2 Basic I/O Concepts](#)
- ▶ [Communicating with Hardware](#)
- ▶ [I/O Systems I/O Hardware Application I/O Interface](#)
- ▶ [COMPUTER ARCHITECTURE](#)
- ▶ [OS](#)
- ▶ [Linux Operating System](#)
- ▶ [W4118 Operating Systems, Instructor: Junfeng Yang](#)
- ▶ [CSNB334 Advanced Operating Systems 4. Process & Resource Management.](#)
- ▶ [Using the POSIX API](#)
- ▶ [Linux Memory Management](#)
- ▶ [Chapter 2: Operating-System Structures](#)
- ▶ [POSIX Threads Programming](#)
- ▶ Pthreads: A shared memory programming model <https://slideplayer.com/slide/8734550/>
- ▶ [Signal Handling in Linux Tushar B. Kute](#)
- ▶ [Introduction to Sockets Programming in C using TCP/IP](#)
- ▶ [How to Run Two or More Terminal Commands at Once in Linux](#)

<https://www.learn-in-depth.com/>
<https://www.facebook.com/groups/embedded.system.KS/>