

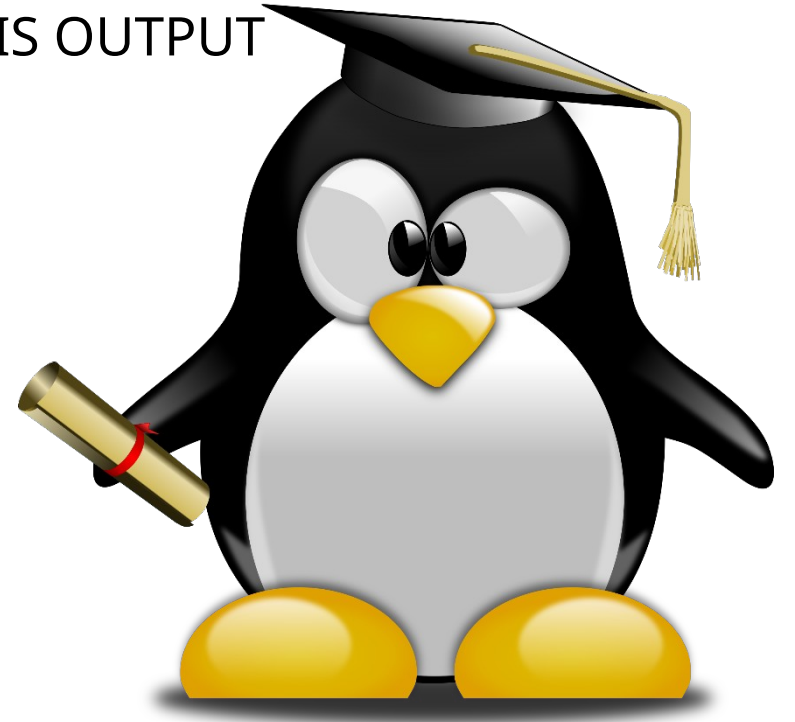


LINUX

AM INTERNSHIP, SEPTEMBER 2018

AGENDA

- HOW TO FIND YOUR FILE
- AND HOW YOU SUPPOSE TO LOOK THROUGH ALL THIS OUTPUT
A GREAT EASE
JUST SEND IT SOMEWHERE ELSE
MAKE. ME. A. SANDWICH.
ALL YOUR FILES ARE BELONG TO US
MATRIOSHKA



HOW TO FIND YOUR FILE

SO YOU CAN GO THIS WAY...

locate <filename>

-i

-c

-n <num>

OR LONG AND CORRECT ONE...

find <Path> <options>

-name

-perm

-user

-size n

-group

-mtime -n + n

-type x

AND HOW YOU SUPPOSE TO LOOK THROUGH ALL THIS OUTPUT

GREP = SEARCH **G**LOBALLY FOR LINES MATCHING
THE **R**EGULAR **E**XPRESSION, AND **P**RINT THEM

```
grep -key 'regex' 'directory'  
grep -key 'regex' file1 file2 ... fileN  
<command> ... | grep 'regex'
```

```
grep -v <template>  
grep -L <template>  
grep -I <template>  
grep -h <template>
```

```
grep -A n <regex> <path>  
grep -B n <regex> <path>  
grep -C n <regex> <path>
```



A GREAT EASE

Symbol	Description	Symbol	Description
!!	Execute the last command	sudo !!	Execute the last command with the sudo permission
!n	Execute the last n line from the history	cd !\$	cd to the directory that was the last argument of the previous command
! <cmd>	Execute the last command that begins from the <cmd>	^ <str1> ^ <str2> ^	Replace the <str1> in the last command with the <str2> and execute
! ?<str>	Execute the last command that contains <str>	! <cmd> :^	The 1 st argument of the command that starts with <cmd>
!!:1	The first argument of the last command	!!:1:r	Remove the extension from the 1 st argument of the last command
!#:^	The first argument of the current command	! <cmd> :<args>:p	Print the command without execution
!:n	The n-th argument of the last command	! <cmd> :x-y	Get the arguments from x to y
!!:1:h	Remove the filename from the 1 st argument of the last command	Ctrl+R	Look for the previous command(same as !?<str>)
!!:1:t	Remove the path from the 1 st argument of the last command		

JUST SEND IT SOMEWHERE ELSE

LETS TALK BOUT STREAMS

`>` - stdout redirect. You also can use `1>`

`>>` - append to a file

`2>` - stderr redirect.

`>|` - allows to rewrite file

`2>&1` / `1>&2` – redirect stdout and stderr to the same file.

`&>` - unites stdout and stderr

`<` or `0<` - redirecting stdin

MAKE. ME. A. SANDWICH.

SUDO = **SUBSTITUTE USER AND DO**

/etc/sudoers – file with sudo rights

sudo -iu <over user>

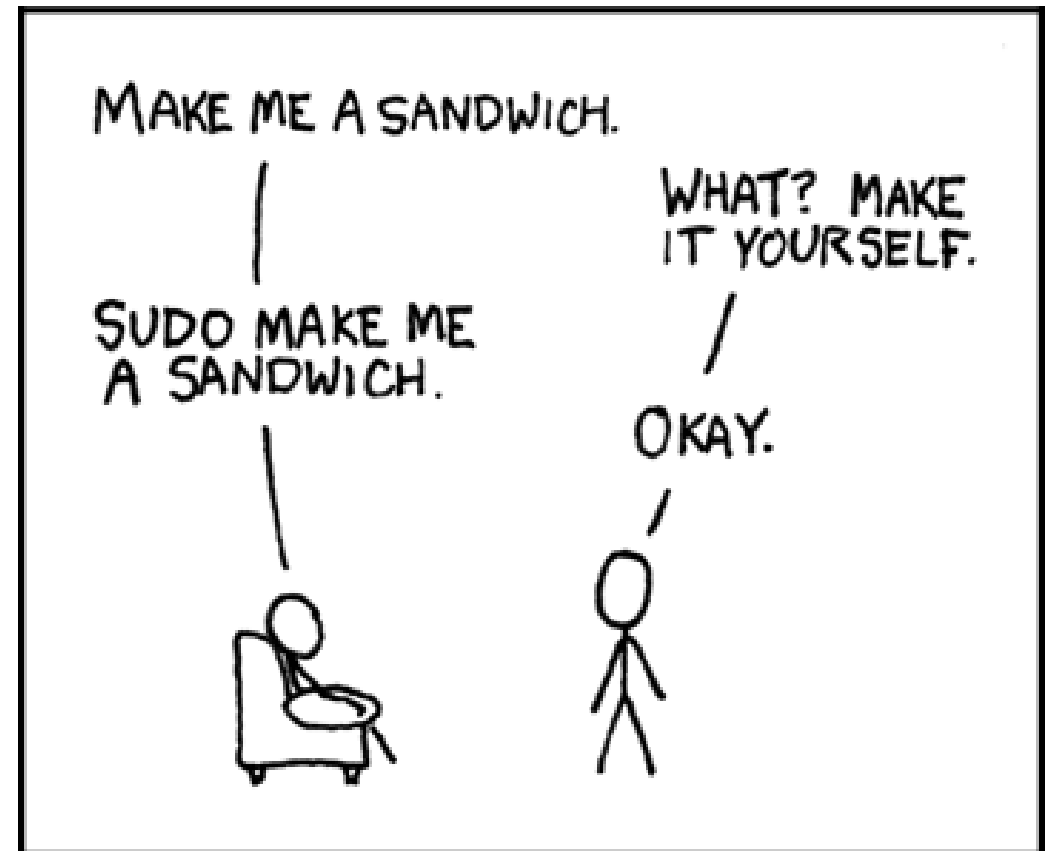
sudo -s -H

sudo -b <command>

sudo -i

sudo -l

sudo -K



ALL YOUR FILES ARE BELONG TO US

OK. SO WHAT'S NEXT ?

File permissions – Read, Write, eXecute

_ rwx rwx rwx

chmod <options> <mode> <file>

-R = recursive

chmod +x <file>

777 – Any user can read, write and execute

400 – Owner can read, Other users cannot do nothing

666 – Any user can read and write

755 – Any user can read and execute, Owner can write

555 – Every user can read and execute with owner rights

chown <options> <new owner> <file>

-R = recursive

MATRIOSHKA

IT`S ALL ABOUT COMPRESSION

gzip <archive name> <files>

gunzip <archive>

-f

-r

-1 --fast

-9 --best

tar -cf - <archive name> <files>

tar -cvzf <archive name> <files>

tar -cvjf <archive name> <files>

tar -xf <archive>

tar -xvf <archive> -C <path>

DIRECTORIES STACK

Add to stack:

`pushd`

Remove from stack:

`popd`

List stack:

`dirs -v`



Linux

DUMITRU GURJUI
DEVOPS ENGINEER