컴퓨터공학설계및실험1

A brief introduction to UNIX

Acknolwedgement

- Some Parts of the lecture notes have been taken from,
  - Bryce Boe, CS32 Summer 2012, UCSB
  - David Mazières, CS140 Operating System, Stanford
  - https://tldp.org/LDP/intro-linux/html/sect_03_01.html
  - https://www.linuxyogi.com/linux-directory-structure-file-system-hierarchy/
  - https://www.gnu.org/software/bash/manual/html_node/Shell-Parameter-Expansion.html
  - Operating System Concepts 9th , Avi Silberschatz
  - https://opensource.com/article/19/4/interprocess-communication-linux-channels

Table of Contents

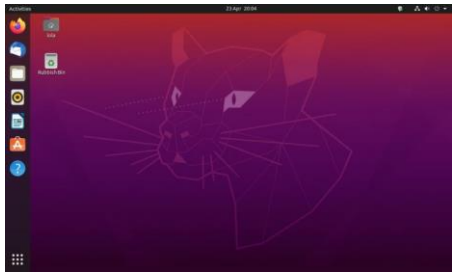
Why use Unix

- Open Source – modifiable, ***free***
- Well-Documented
- Community Support
- Great Stability

Windows Server 2022 Edition	Ideal for	Licensing model	CAL requirements [1]	Suggested Retail Price (MSRP) [4]
Datacenter [2]	Highly virtualized datacenters and cloud environments	Core-based	Windows Server CAL	$6,155
Standard [2]	Physical or minimally virtualized environments	Core-based	Windows Server CAL	$1069
Essentials	Small businesses with up to 25 users and 50 devices	Specialty servers (server license) [3]	No CAL required	$501

Unix-like OS

- ❖ Almost follows Unix standard specifications:
  - ❖ Process management, C Library, I/O control …
- Linux
  - Popular distributions: Ubuntu, Manjaro, …
  - Unique configuration of desktop environment, third party apps(office, web-browser, video player, …), package management system(apt, pacman, aur, …)
- macOS

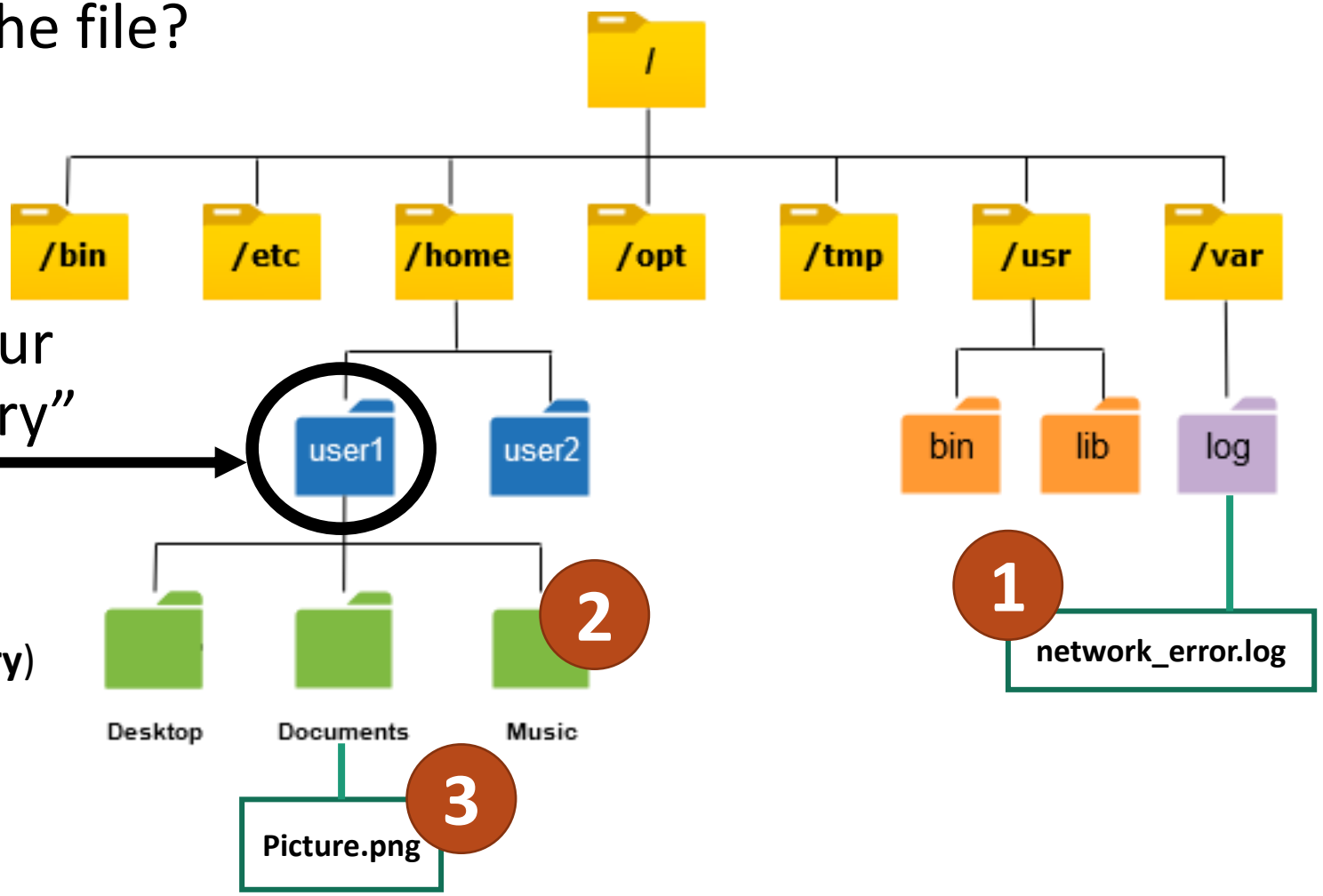


Unix 1) File

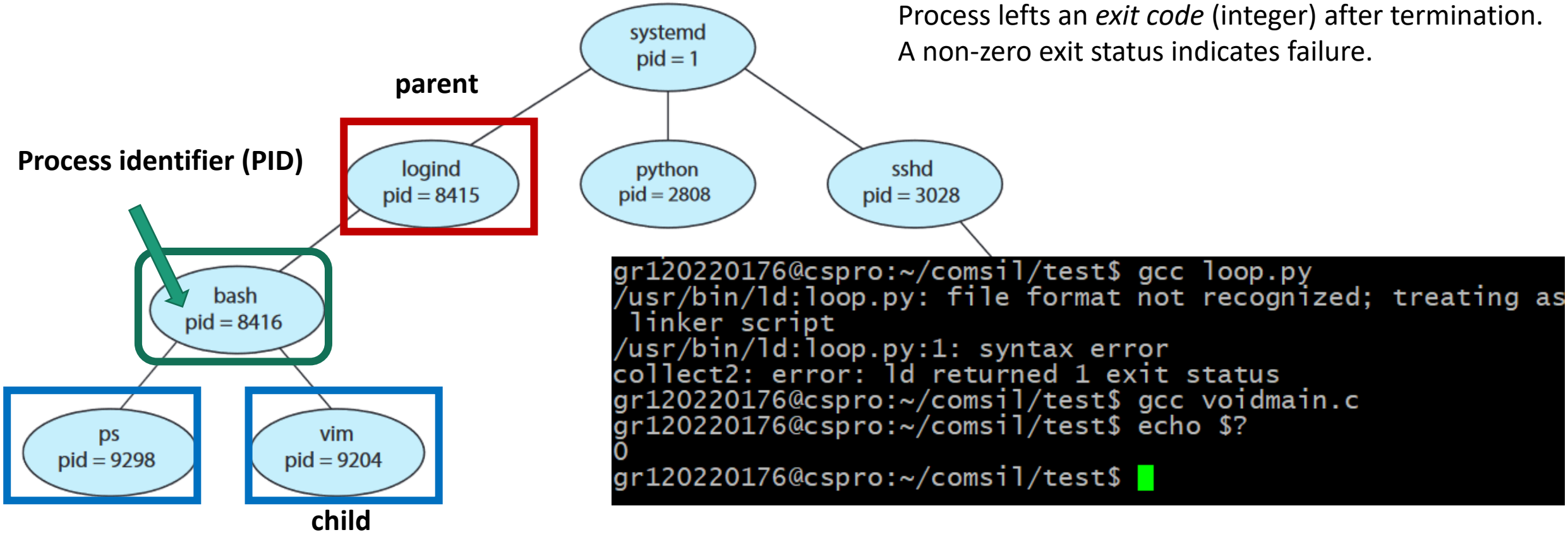
- How can a process locate the file?
  - Absolute Path
  - Relative Path

- Locate the file based on your "(Current) Working directory"

1. ../../var/log/network_error.log
   /var/log/network_error.log (abs)
2. ./Music
   ~/Music (based on your **home directory**)
   /home/user1/Music
3. ./Document/picture.png



Unix 1) Process



Process lefts an *exit code* (integer) after termination.
A non-zero exit status indicates failure.

Figure 3.7 A tree of processes on a typical Linux system.

Unix 1) Process

- How can a process interact with its environment(user, process, …)?
  - **Argument passing**
  - Through Files
- You can pass arguments to the program by appending the command line at the end. The line is split into tokens by space(‘ ‘). After that, Expansion can be performed (usually with the combination of regex and shell’s variables)
  https://www.gnu.org/software/bash/manual/html_node/Shell-Expansions.html

```c
#include <stdio.h>

int main(int argc, char* argv[]) {
  for (int i = 0; i < argc; i++)
    printf("%s\n", argv[i]);
  return 0;
}
```

- args.c -

alpaca@cspro:~$ gcc -o show_args args.c

alpaca@cspro:~$ ./show_args arg1 arg2

./show_args
arg1
arg2

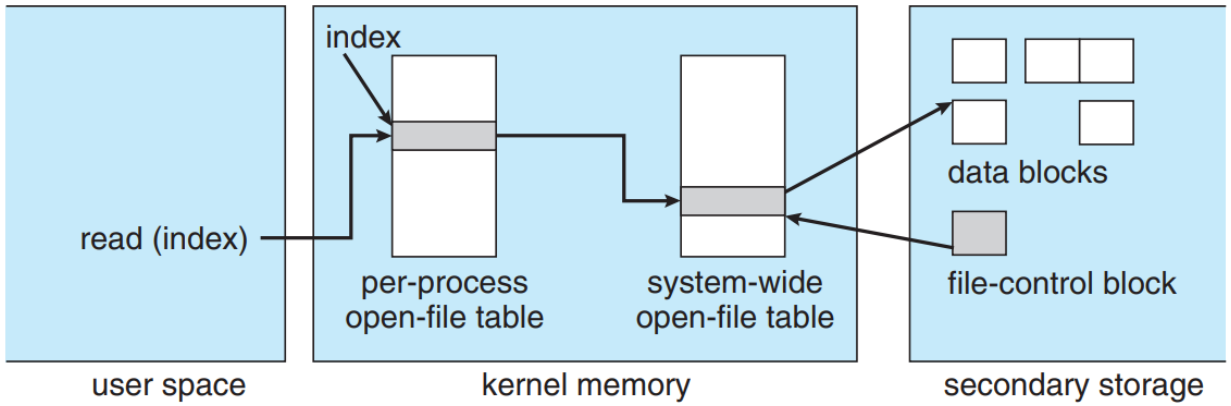
0 1 2

cse20234567@cspro:~$ ./show_args *

→ ***create some files at the same directory and try yourself!***

Unix 1) Process

- How can a process interact with its environment(user, process, …)?
  - Argument passing
  - **Through files**
- Physical devices like keyboard, monitor, printer, scanner … are abstracted in form of ‘file’. A process can interact with them in the manner of [open → read / write] to the file.
- Each process can access opened files using a unique index(unsigned integer) called as ‘file descriptor’.



Unix 1) Process

writing msg to stdout

```
$ echo “test” > temp
$ cat temp
test
$ echo “test” > /dev/stdout
test
```

```c
#include <stdio.h>

int main() {
    int x = 2023;
    printf("fd of stdout = %d\n", fileno(stdout));
    fprintf(stdout, "%d\n", x);
    FILE* fp = fopen("temp", "w");
    printf("fd of temp = %d\n", fileno(fp));
    fprintf(fp, "%d\n", x);
    fclose(fp);
    return 0;
}
```

========== output ==========
fd of stdout = 1
2023
fd of tmp = 3

Unix 2) Redirection & Anonymous Pipe

- We don’t need to type all the inputs by keyboard.
  - Solution 1: Redirection
    The shell interprets the redirection symbols <,>, and >> as instructions to reroute a command’s STDIN(=from keyboard) or STDOUT(=to screen of shell) to or from a file.
  - Solution 2: Pipe
    To connect the STDOUT of one command to the STDIN of another, use the | symbol, commonly known as a pipe.
  - What are the differences?

Unix 2) Redirection & Anonymous Pipe

```c
// readints.c
#include <stdio.h>

int main() {
    printf("read ints\n");
    for (int i = 1; i <= 5; i++) {
        int x;
        scanf("%d", &x);
        printf("int %d: %d\n", i, x);
    }
    return 0;
}
```

```c
// printints.c
#include <stdio.h>
#include <unistd.h>

int main() {
    for (int i = 1; i <= 5; i++) {
        printf("%d\n", i);
        fflush(stdout);
        sleep(1);
    }
    return 0;
}
```

- Compile and try yourself.

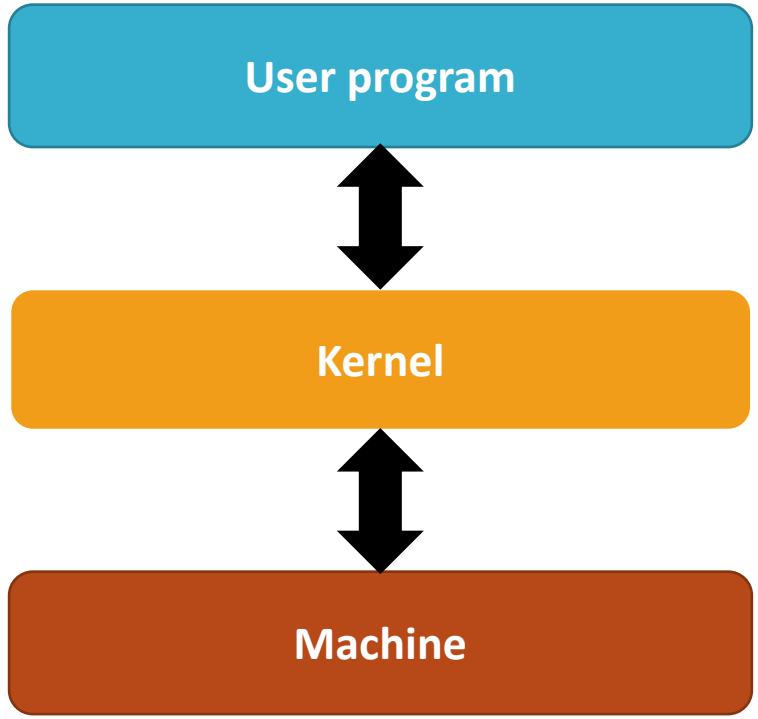
```
./printints > temp
./readints < temp
```

```
./printints | ./readints
```

Unix 3) Kernel & Shell

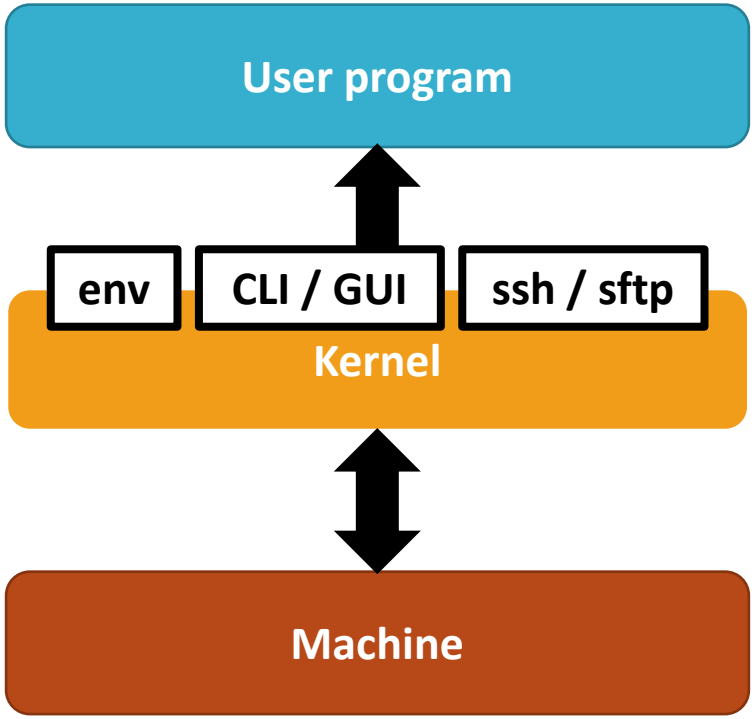
- The kernel provides applications with system services such as I/O management, memory management and scheduling.

- *This can be a whole architecture for embedded / real-time system if the limited number of programs run on the machine.*



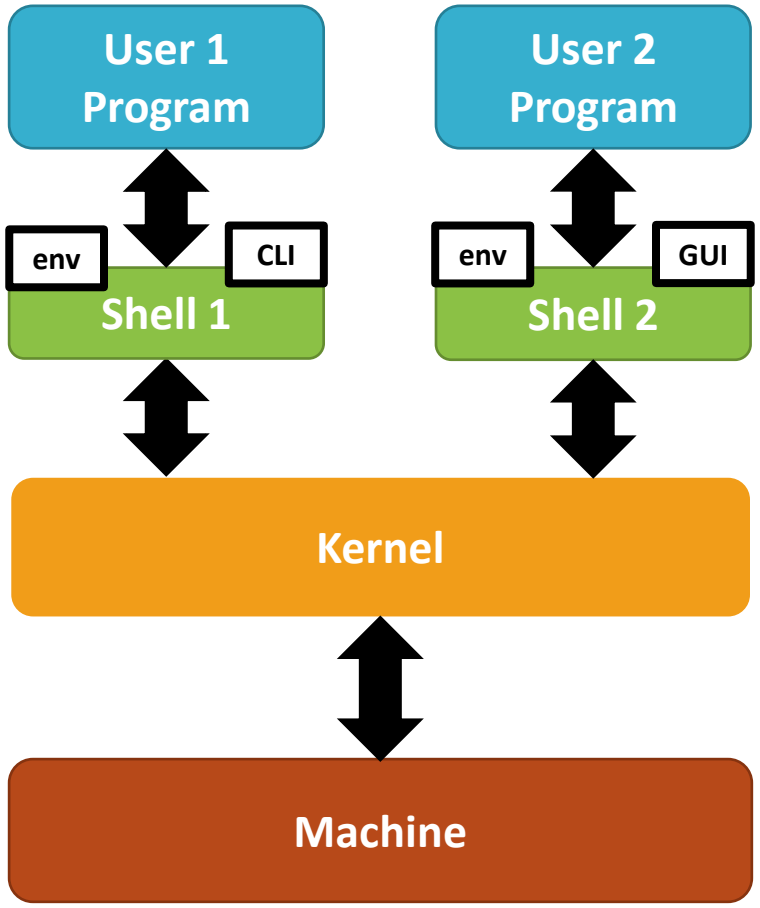
Unix 3) Kernel & Shell

- User program may share some common options / configuration like language settings, user-name, location of home directory, …
- Let's treat them as **env**ironment variables, and…



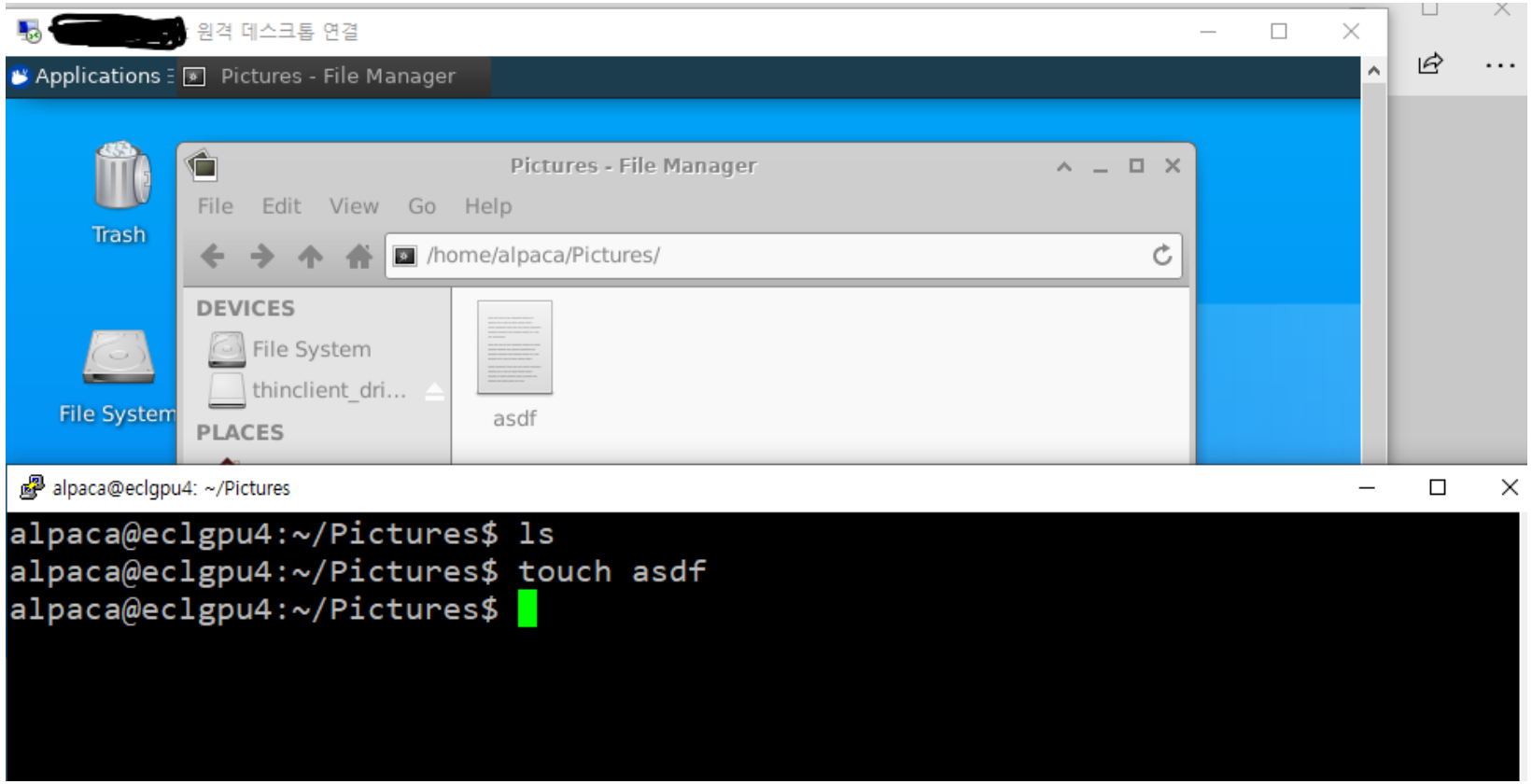
Unix 3) Kernel & Shell

- User program may share some common options / configuration like language settings, user-name, location of home directory, …
- Let's treat them as **env**ironment variables, and… separate them as an independent module called ***shell!***



Unix 3) Kernel & Shell

Example: Accessing the same linux server with different shells

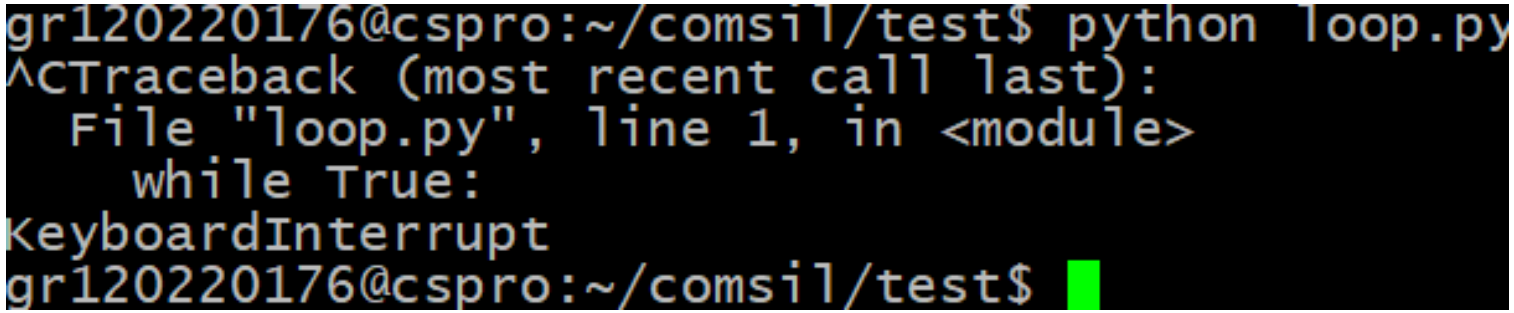


xfce (GUI)

Bash (CLI)

Unix 3) Kernel & Shell

- Shell provides
  - Process creation
  - Standard Input / Output
  - Signal handling
    ex) A program is killed when you press *ctrl + c* by default.



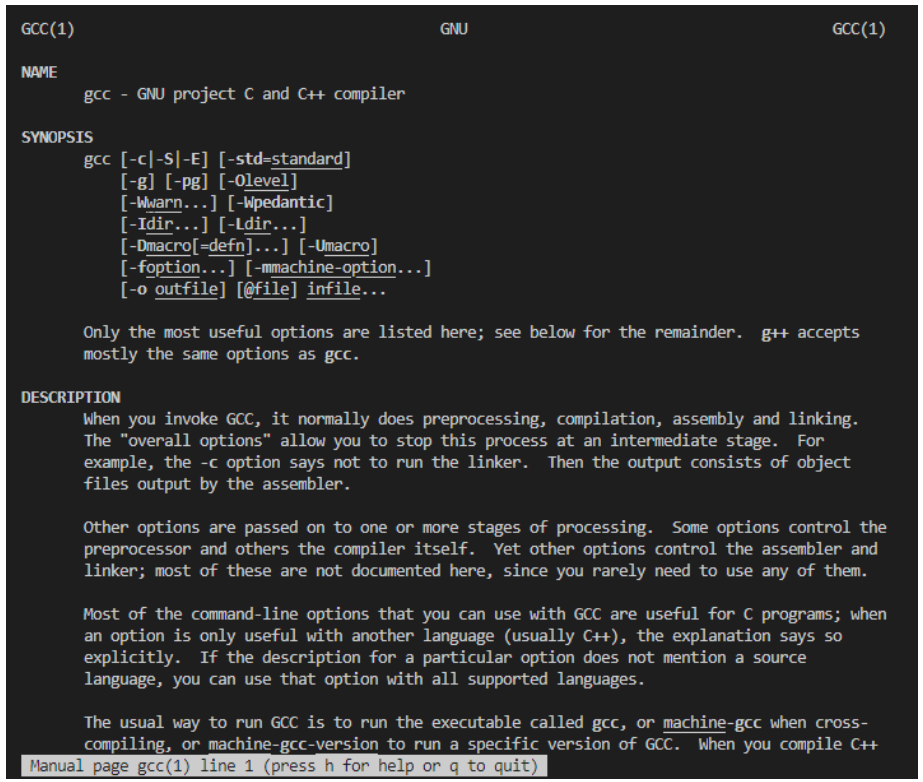
Shell Programming

- GUI is handy and very intuitive to use even if a baby can use.
- But for advanced users, it isn't powerful yet…
  - Find the total size of images within a specific folder
  - Report the network status for every 10 seconds
  - Create more than hundreds of accounts (cspro!)

- It is difficult to write your shell program for the first time. Try to…
  - *Split the task into sub-tasks and reassemble them*
  - *Read the error messages and struggle to understand as far as you can*

Shell Programming - manual

1. man [command]

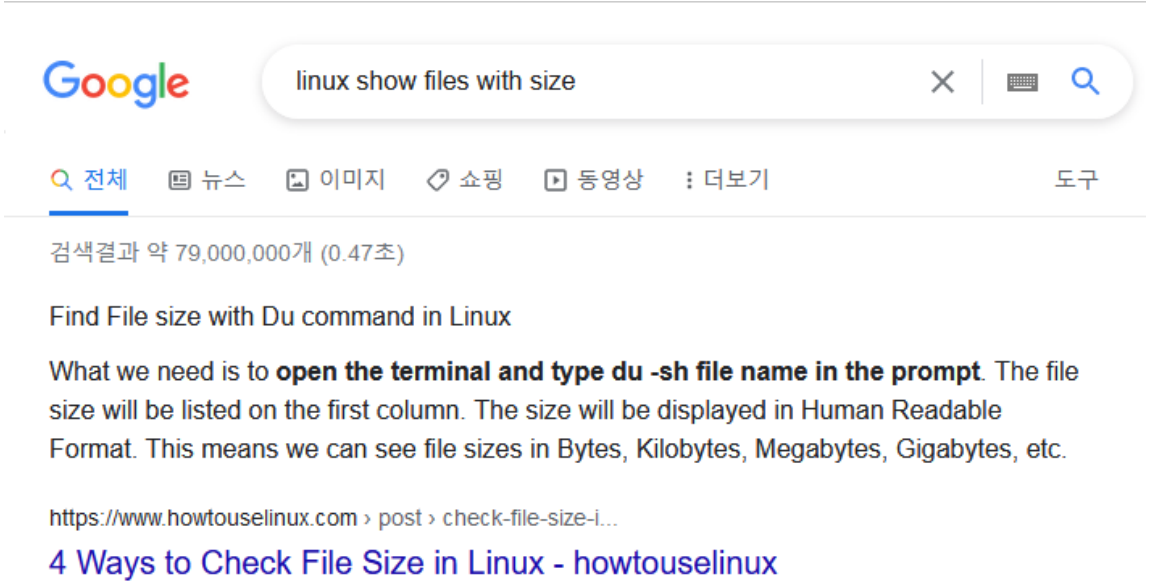


```
GCC(1)                                  GNU                                  GCC(1)

NAME
       gcc - GNU project C and C++ compiler

SYNOPSIS
       gcc [-c|-S|-E] [-std=standard]
           [-g] [-pg] [-Olevel]
           [-Wwarn...] [-Wpedantic]
           [-Idir...] [-Ldir...]
           [-Dmacro[=defn]...] [-Umacro]
           [-foption...] [-mmachine-option...]
           [-o outfile] [@file] infile...

       Only the most useful options are listed here; see below for the remainder.  g++ accepts
       mostly the same options as gcc.

DESCRIPTION
       When you invoke GCC, it normally does preprocessing, compilation, assembly and linking.
       The "overall options" allow you to stop this process at an intermediate stage.  For
       example, the -c option says not to run the linker.  Then the output consists of object
       files output by the assembler.

       Other options are passed on to one or more stages of processing.  Some options control the
       preprocessor and others the compiler itself.  Yet other options control the assembler and
       linker; most of these are not documented here, since you rarely need to use any of them.

       Most of the command-line options that you can use with GCC are useful for C programs; when
       an option is only useful with another language (usually C++), the explanation says so
       explicitly.  If the description for a particular option does not mention a source
       language, you can use that option with all supported languages.

       The usual way to run GCC is to run the executable called gcc, or machine-gcc when cross-
       compiling, or machine-gcc-version to run a specific version of GCC.  When you compile C++
Manual page gcc(1) line 1 (press h for help or q to quit)
```

$ man gcc

2. Google

- Search in English
- Specify the keywords



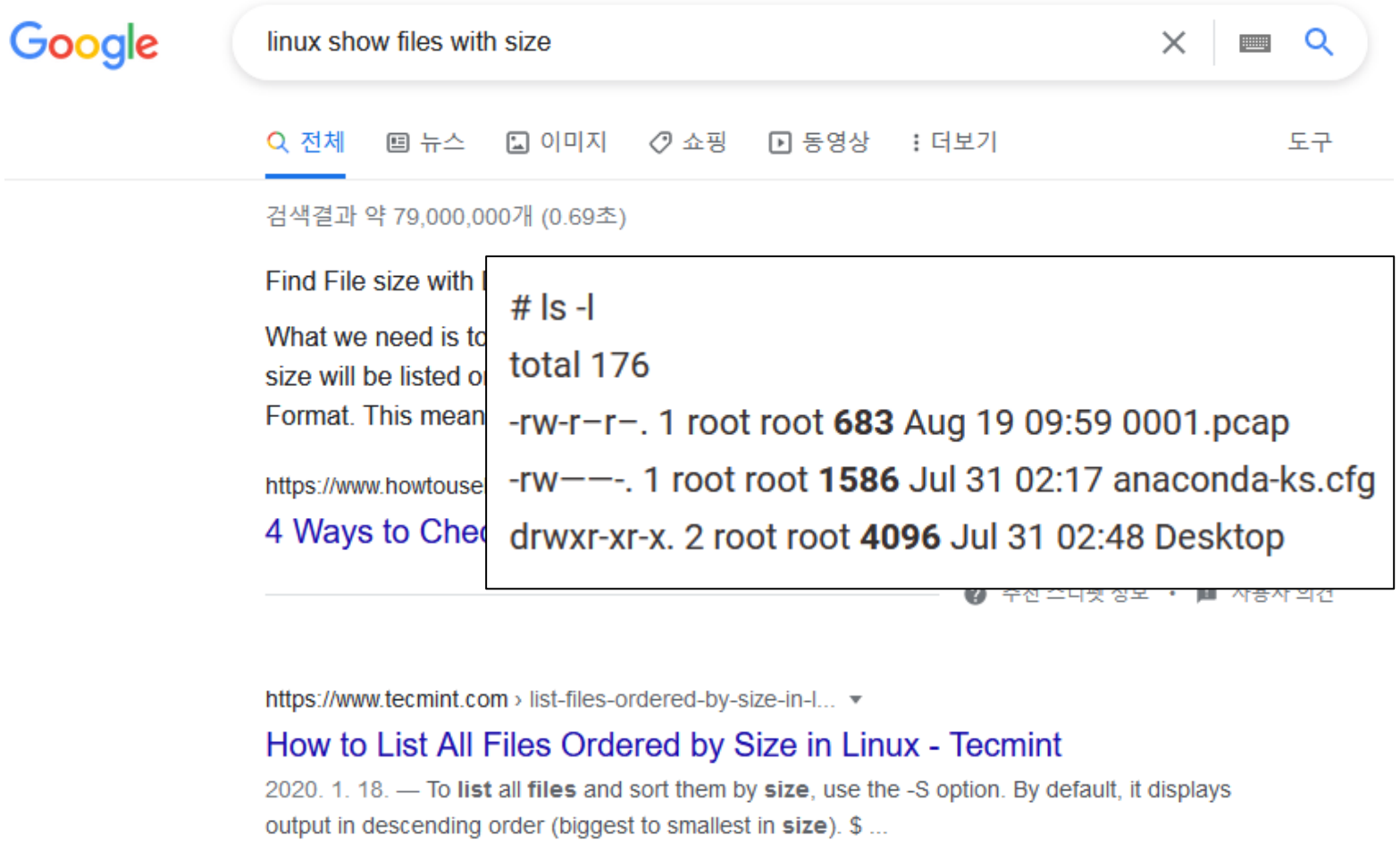
Shell Programming - regex

- Pattern matching Language, extremely useful when filtering the output
- Try yourself
  - https://regexone.com/
  - https://regexlearn.com/learn/regex101

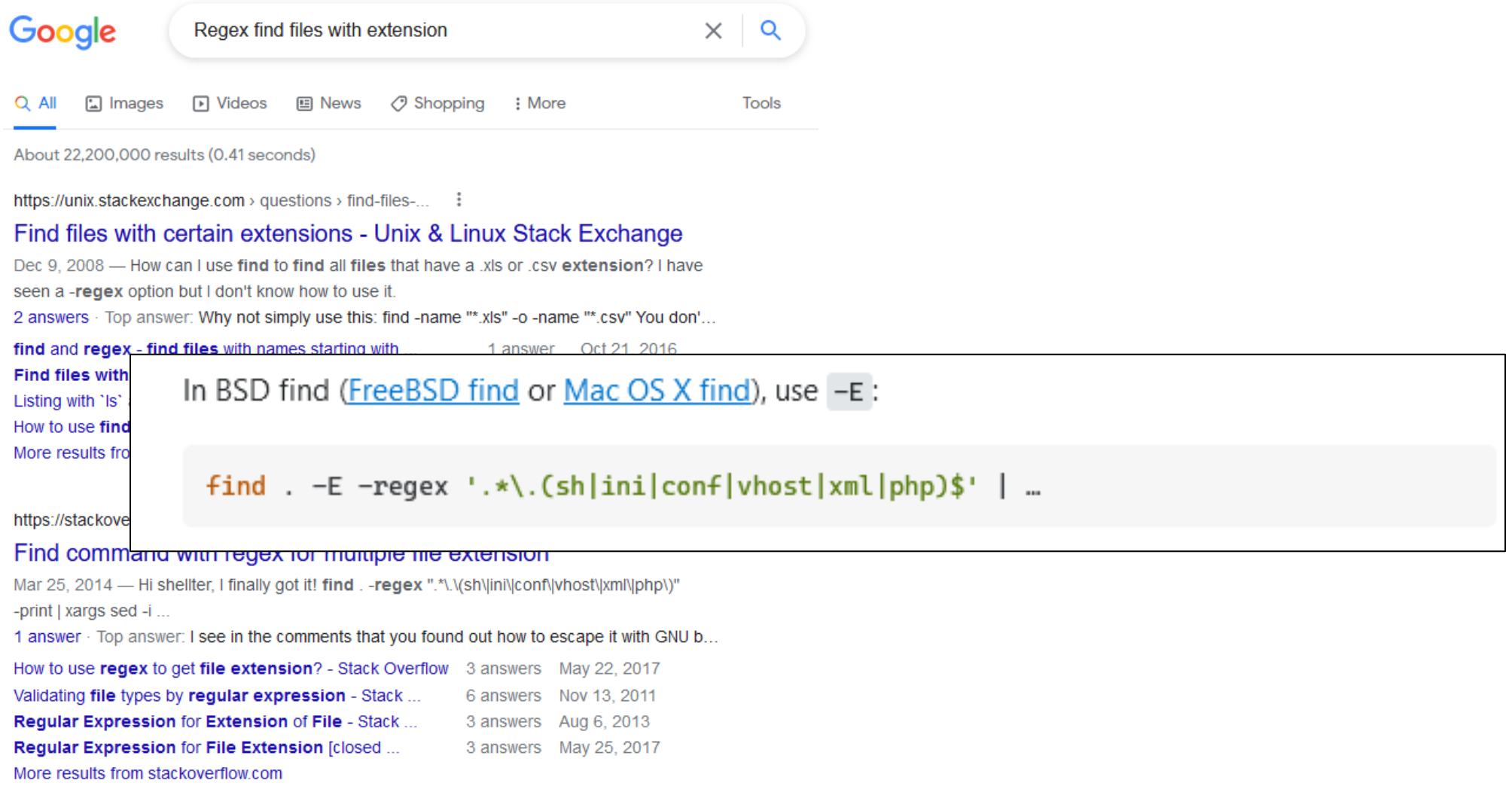
Shell Programming - showcase

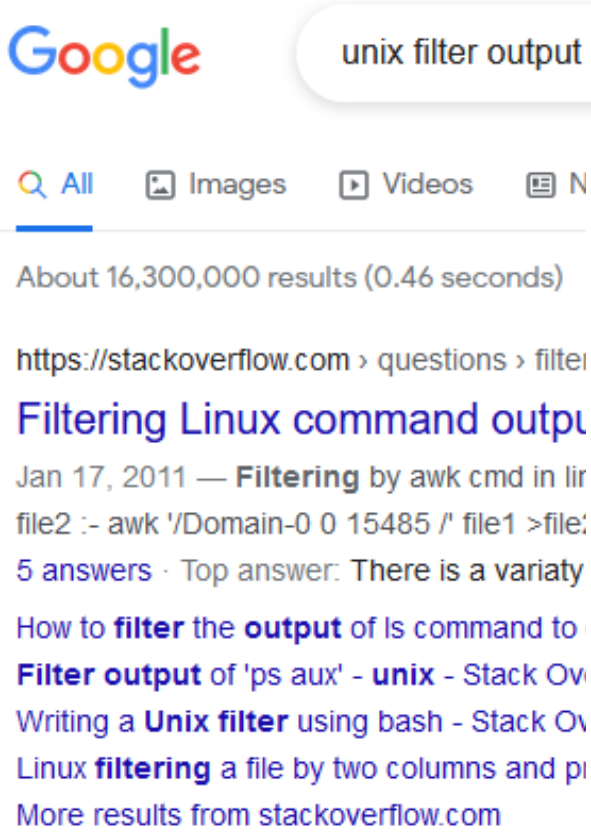
- Find the total size of images within a specific folder
  1. Get the size of a single file
  2. List the files that ends with jpg, png, gif, …
  3. Sum up the size of individual files

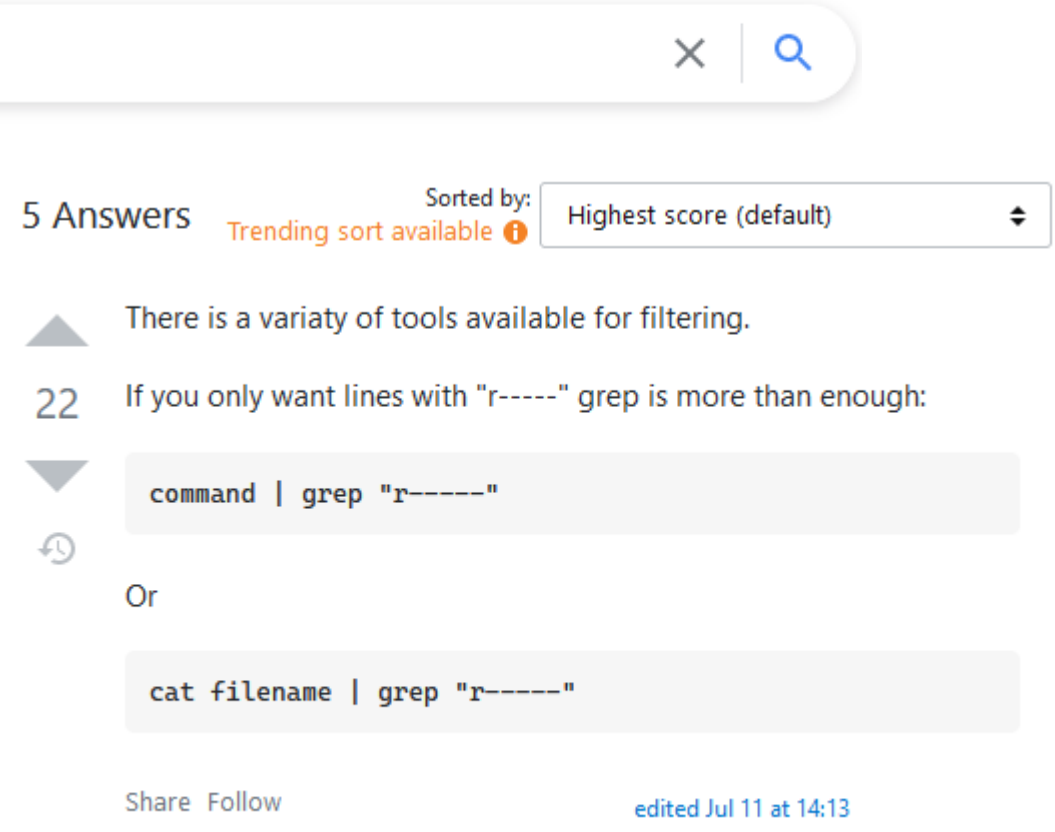
Shell Programming - showcase



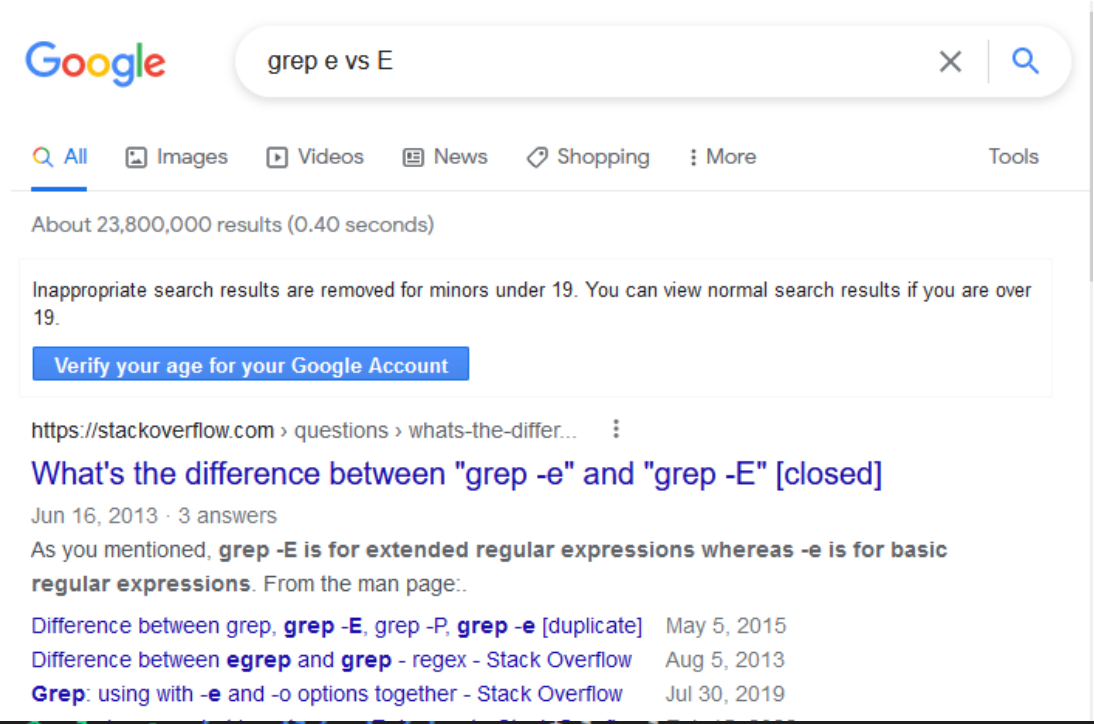
Shell Programming - showcase




Google
unix filter output
All
Images
Videos
N
About 16,300,000 results (0.46 seconds)
https://stackoverflow.com › questions › filter
Filtering Linux command outpu
Jan 17, 2011 — Filtering by awk cmd in lir
file2 :- awk '/Domain-0 0 15485 /' file1 >file2
5 answers · Top answer: There is a variaty
How to filter the output of ls command to
Filter output of 'ps aux' - unix - Stack Ov
Writing a Unix filter using bash - Stack Ov
Linux filtering a file by two columns and pr
More results from stackoverflow.com


5 Answers
Sorted by:
Trending sort available
Highest score (default)
22
There is a variaty of tools available for filtering.
If you only want lines with "r-----" grep is more than enough:
command | grep "r-----"
Or
cat filename | grep "r-----"
Share
Follow
edited Jul 11 at 14:13

```
GREP(1)                          User Commands                          GREP(1)

NAME
       grep, egrep, fgrep, rgrep - print lines that match patterns

SYNOPSIS
       grep [OPTION...] PATTERNS [FILE...]
       grep [OPTION...] -e PATTERNS ... [FILE...]
       grep [OPTION...] -f PATTERN_FILE ... [FILE...]

DESCRIPTION
       grep  searches  for  PATTERNS in each FILE.  PATTERNS is one or more patterns separated by
       newline characters, and grep prints each line that matches a pattern.  Typically  PATTERNS
       should be quoted when grep is used in a shell command.

       A  FILE of "-" stands for standard input.  If no FILE is given, recursive searches examine
       the working directory, and nonrecursive searches read standard input.

       In addition, the variant programs egrep, fgrep and rgrep are the same as grep -E, grep -F,
       and  grep -r,  respectively.  These variants are deprecated, but are provided for backward
       compatibility.

OPTIONS
   Generic Program Information
       --help Output a usage message and exit.

       -V, --version
              Output the version number of grep and exit.

   Pattern Syntax
       -E, --extended-regexp
              Interpret PATTERNS as extended regular expressions (EREs, see below).

       -F, --fixed-strings
Manual page grep(1) line 1 (press h for help or q to quit)
```



- `ls -l | grep -E '.*\.(png|jpg|gif)$'`

```
alpaca@eclgpu4:~/comsil/unix/showcase$ ls -l
total 16
-rw-rw-r-- 1 alpaca alpaca  9 Sep 12 10:59 apple.png
-rw-rw-r-- 1 alpaca alpaca 20 Sep 12 10:59 banana.png
-rwxrw-r-- 1 alpaca alpaca 37 Sep 12 10:52 go.sh
-rw-rw-r-- 1 alpaca alpaca 11 Sep 12 11:00 kiwi.jpg
-rw-rw-r-- 1 alpaca alpaca  0 Sep 12 10:59 memo.txt
-rw-rw-r-- 1 alpaca alpaca  0 Sep 12 10:59 page.html
-rw-rw-r-- 1 alpaca alpaca  0 Sep 12 10:59 snake.py
alpaca@eclgpu4:~/comsil/unix/showcase$ ls -l | grep -E '.*(png|jpg|gif)$'
-rw-rw-r-- 1 alpaca alpaca  9 Sep 12 10:59 apple.png
-rw-rw-r-- 1 alpaca alpaca 20 Sep 12 10:59 banana.png
-rw-rw-r-- 1 alpaca alpaca 11 Sep 12 11:00 kiwi.jpg
alpaca@eclgpu4:~/comsil/unix/showcase$
```

Google

bash select specific column

All Videos Images Shopping News

About 19,900,000 results (0.80 seconds)

https://stackoverflow.com › questions › how-to-get-the-...

How to get the second column from command out

Apr 21, 2013 — The first **column** is always a number, followed by a sp
quoted string. My purpose is to **get** the second **column** only, like: "A
8 answers · Top answer: Use -F [field separator] to split the lines on "s

Bash Extract **specific columns** from TSV file to new file and ... Mar
selecting specific column on **linux** terminal - Stack Overflow Dec
How to print **specific columns** from file using **bash**? Oct
while read line in a file from a **specific column** [duplicate] Jun
More results from stackoverflow.com

https://unix.stackexchange.com › questions › how-to-ex...

how to extract certain columns - Unix & Linux Sta...

May 10, 2017 — I have a data set and I need to extract **certain columns** using **linux** for example. I have the following **columns**.
4 answers · Top answer: Your question is lacking a bit of details. I assume it is something fou...

How to use a shell command to only show the first ... 6 answers Mar 7, 2017

8 Answers

Sorted by: Highest score (default)

Trending sort available

267

Use `-F [field separator]` to split the lines on `"`s:

```
awk -F '"' '{print $2}' your_input_file
```

or for input from pipe

```
<some_command> | awk -F '"' '{print $2}'
```

output:

```
A B
C
D
```

Share Follow

Google

bash sum up integers

All Images Videos News Sh

Plain bash:

91

```
$ cat numbers.txt
1
2
3
4
5
6
7
8
9
10
$ sum=0; while read num; do ((sum += num)); done < numbers.txt;
55
```

comsil > unix > showcase > $ go.sh

```
1  ls -l | grep -E '.*\.(png|jpg|gif)$' | awk -F ' ' '{print $5}' > temp
2  sum=0
3  while read num; do ((sum += num)); done < temp
4  echo $sum
5  rm temp
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPYTER bash - showcase

```
alpaca@eclgpu4:~/comsil/unix/showcase$ ./go.sh
40
alpaca@eclgpu4:~/comsil/unix/showcase$
```