# Spring 2022 courses

## `DONE` os Test 2

### SETTINGS

Operating Systems (CSC 420) test 2:

- 10 questions from quiz 4-6 (some questions modified)
- 10 new questions
- Partial credit enabled where applicable
- You can change answers before submitting
- You can resume an incomplete submission
- It is mentioned if > 1 answer is correct

Good luck!

### How can you start Emacs on the command line and continue working in that terminal window?

Tip: usually, when you enter only the command `emacs` at the command prompt, that terminal window is busy with this one process.

TRUE:

- `emacs &`

FALSE:

- That only works on a Mac
- `emacs -nw`
- `emacs #`

### As user `pi`, in which directory is your `$HOME` directory (in the Raspbian OS)?

TRUE:

- /home/pi

FALSE:

- /usr/pi
- c:/User/pi
- /root/pi

### When you type `sudo` before a command, you become the `root` user

FALSE

Tip: `sudo` stands for `Super User DO` and is used to access restricted files and operations like system-wide installations

### What's wrong with this command?

All of these commands generate the same error "command not found".

What am I doing wrong?

```
ls-lt text.csv
```

TRUE:n

- The command is ls, not ls-lt, -lt is the option

FALSE:

- text.csv does not exist
- -lt is not a valid option for ls
- ls does not work on .csv files

## Match file descriptor and OS communication stream

| | |
|---|---|
| standard input | 0 |
| standard output | 1 |
| standard error | 2 |

## Complete the command to get the output

Replace `???` in the code block by the correct expression to get the results shown below.

```
ls | ??? org
```

RESULT:

```
booklist.org
todo.org
todo.org_archive
```

TRUE:

- grep

FALSE

- cut
- tee
- head

## What is the output of this command sequence?

```
mkdir dir1 dir2
echo "found: " *[0-9]
```

TRUE:

- found: dir1 dir2

FALSE:

- "found: " *[0-9]
- mkdir: created directory 'dir1' 'dir2'

- command not found

## Complete the code to get the output

How can you create a symbolic (soft) link from the file `foo` to a link called `foo-sym`? Replace [1] and [2] in the code block to get the results shown below.

```
[1] -s [2] [3]
```

TRUE:

- [1] ln [2] foo [3] foo-sym

FALSE:

- [1] ls [2] foo [3] foo-sym
- [1] ln [2] foo-sym [3] foo
- [1] ln [2] | [3] foo-sym

## Complete the code to save intermediate output

In the following command, we count the number of ZIP files in the directories `/bin` and `/usr/bin` using a pipeline:

- `ls` lists the files
- `grep` searches for the zip pattern in the file names
- `wc -l` counts the output lines (one for each zip file found)

  Replace `???` in the code block by the expression to save a copy of the file listing in the file `ls.txt` without interrupting the pipeline!

  ```
  ls /bin /usr/bin | ??? ls.txt | grep zip | wc -l
  ```

  TRUE:

  - tee

  FALSE:

  - cp
  - uniq
  - sort

## Redirect and append stdout and stderr to an output file

You can use the operator `&>` to **redirect** both standard error and standard output to one and the same file `out.txt`. But this file will be overwritten every time we use the operator.

Which operator would you use to **append** standard error and standard output so that no messages are lost?

*More than answer is correct.*

TRUE:

- ls -l /etc/passwd &>> out.txt
- ls -l /etc/passwd >> out.txt 2>&1

FALSE:

- ls -l /etc/passwd | tee out.txt
- ls -l /etc/passwd >> out.txt

## 6 How can you create an alternate command?

When using the command line, you find that you always type `ls -la` - you always want to see all files, including "hidden" dot files, in long format. How can you define a command `ll` that always does exactly that, execute `ls -la`?

TRUE:

- `alias ll='ls -la'`

FALSE

- `ll <- ls -la`
- Not possible, `ls` cannot be redefined
- You have to write a `bash` script to do that

Feedback: you can create an alias for a simple or a compound command (like an entire pipeline) with the `alias` command. An alias can be unmade with the command `unalias`. The `alias` command creates a permanent command alias. That's why the command `unalias` exists. You can find all aliases by typing `alias` at the prompt. You can even override standard commands. E.g. if you always want `ls` to show all files in long format, alias it to `ls -la`. However, for standard utilities, the OS might override your choices at boot time. If you try this at home: you cannot have spaces between the alias name, the = and the definition string!

## The pipe (`|`) lets you use two or more commands such that output of one command serves as input to the next.

TRUE

## 6 Be the system!

Which of the statements below about the file listing are true?

```
71776119061327028 lrwxrwxrwx 1  8 Mar  5 16:34 hello -> hello.sh
25614222880787780 -rw-r--r-- 2 34 Jan 25 08:29 hello.sh
25614222880787780 -rw-r--r-- 2 34 Jan 25 08:29 hello_h
```

*More than one answer is correct.*

TRUE:

- `hello` is a symbolic link
- The listing shows the file `inodes`
- `hello_h` is a hard link of `hello.sh`

FALSE:

- `hello` is a broken link

## 5 CHANGED Be the shell

The text file `text.csv` has 10 columns and 100 lines. Select columns 1 and 2, and print out the number of words in the first 10 lines of these columns.

TRUE:

- `cut -f1-2 -d , text.csv | head -n 10 | wc -w`

FALSE:

- `cut -f1-2 -d , text | head -n 10 | wc -l`
- `cut -f1,2 -d , text.csv | wc -w | head -n 10`
- `cut -f1+2 -d , text.csv | head -n 10 | wc -w`

Feedback: the shell pipeline goes from left to right. Each part follows the recipe [command] [options] [file]. The column selection command is `cut`. To select columns, use `-f` followed by the column numbers. Followed by `|` redirects the output (stdout) of the first operation into input (stdin) of the second operation. This is `head`, which filters `-n` lines from its input. Finally, another pipe leads to `wc` which counts the words with `-w`.

## 5 CHANGED On the shell, the order of command and option is relevant.

That is, `-n 10 head file` gives an error because `-n` is not recognized.

TRUE

Feedback: The order is significant because the shell operates a pipeline from left to right: if the first characters after the prompt are `-n` then the shell will look for (and not find) a program called `-n`. The order of the options themselves in the option block (between command and target) is also not irrelevant: e.g. `rm -if file` removes file, while `rm -fi file` asks for confirmation!

## 5 What is "tab completion" on the shell?

*More than one answer possible.*

TRUE:

- The shell tries to auto-complete the path to a file
- Emacs tries to auto-complete commands after `M-x`

FALSE:

- The shell looks through all files to find the file you want
- When pressing the TAB key, an invisible TAB character is inserted

## 5 What is the relationship between the shell and the operating system (OS)?

*More than one answer possible.*

TRUE:

- The shell exposes OS services to humans or other programs
- The shell gives OS access through a command-line (CLI) or a graphical user interface (GUI)

FALSE:

- The shell is part of the OS kernel program
- The shell contains the OS kernel program

## 4 Which of these statements about shell commands are NOT true?

TRUE:

- Commands are not case-sensitive

- Shell commands must be entered on the terminal to be executed

FALSE:

- Shell commands have the structure [cmd] -[options] [arguments]
- Long option have two dashes, like `--reverse` for `ls`

Feedback: Shell commands are case-sensitive, as are all names. This is not so on Windows (e.g. `DIR` and `dir` on the CMD line terminal). Shell commands can be put in `.sh` files and run by the shell program (e.g. `bash`).

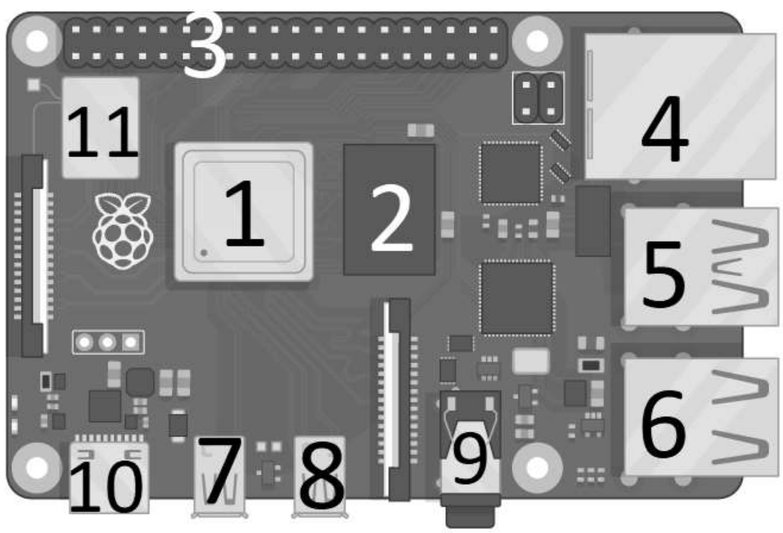
## 4 In Linux, you can have spaces in file names, like `hello world.exe`

FALSE

Feedback `.exe` is a file ending for an executable file on Windows where spaces in file names are frequent (and frequently annoying). On Linux, and in all programming languages, spaces are replaced by underscores.

## 4 Raspberry Pi 4 Architecture

Name the parts of the Raspberry Pi 4 board in the picture that are missing in the list:



1. Processor (SoC - System on a Chip)
2. Memory (RAM)
3. General Purpose Input Output (GPIO) pins
4. Ethernet port

---

5 USB-3 ports

6 USB-2 ports

7 Micro HDMI display output 1

8 Micro HDMI display output 2

9 Jack 3.5mm output (earphone)

10 USB-C power input

## 11 Wireless card

Author: Marcus Birkenkrahe

Created: 2022-03-17 Thu 15:24

Validate