University of Reading
Department of Computer Science

Julian Kunkel

Exercise week 1
CS3DP / Autumn 2019
30 Marks Total
Due Date: 2019-10-07

---

1. The tasks described in this worksheet are part of the formative assessment, they serve the purpose to prepare you for the examination. We will discuss the solutions during the tutorial/exercise session on the next Monday they are handed out.

2. Make sure to carefully plan your time for the whole sheet, the complete exercise should represent approximately 5 hours of independent study. Each mark attained should correspond to roughly 10 minutes of your time and includes the time to check the lecture notes. The time limit might be too ambitious for you. It indicates how much time you should spent on each task not how much time you may actually need.
You should get it at least partially resolved with the time budget, if you are struggling then reach out for help at `http://cs-reading.slack.com`. Join here.

3. We recommend, that you create a (private) GIT repository where you store your findings and outcomes while processing the exercises. This portfolio of work could be useful in the future.

Contents

Task 1: Understanding System Performance (9 Marks)

Performance is an important aspect of parallel and distributed systems. In this task, you will research relevant performance characteristics of a single computer system and a distributed system using literature.

Tasks

1. Research what the term "performance" could mean for a single computer system. Note that there is not the single performance characteristics but multiple characteristics that are meaningful.

2. Document relevant characteristics single computer system, i.e., a server or desktop system. Find typical values for a decent (current) computer system in the document. The notes could cover about 1/2-page.

3. Think about performance of multiple computers that are interconnected to a distributed system. How would you expand upon your performance characteristics?

Submission (directory: `1/performance`)

`1/performance/fb.txt`	The feedback file for this exercise.
`1/performance/performance-1node.txt`	Performance characteristics of a single node.
`1/performance/performance-distributed.txt`	Performance characteristics of a distributed system.

Task 2: Bash Exercises (9 Marks)

Supercomputers and distributed computers often use Linux to interact with them. This exercise aids to provide an introduction to bash programming, in case you are in need.

We don't expect for you to be writing long and complex bash scripts straight away. We also appreciate that moving to a CLI (Command Line Interface), while powerful, can be challenging especially when you are accustomed to interacting with primarily graphical interfaces on Operating Systems like Windows.
Below we have included several Bash commands of varying complexity that also cover more advanced concepts of the Bash. We want you to experiment with them, try running them and observe the result! Think about what you expect their output to be before you run them. Perhaps make your own changes to the commands. Use this as an opportunity to learn about the shell and increase your comfort in writing your own scripts.
We would like you to produce a text file providing a short descriptions of your findings, i.e. all you need to provide is a brief description of what the command does (if possible write what you expected) and a short explanation of how it works where appropriate.

Run the following command **BEFORE** beginning the exercise, this command will create a directory and change your current directory to it:
`$ mkdir -p $HOME/exercise1 && cd $HOME/exercise1`
You should complete the steps below in this directory.

1. `$ echo "Hello World"`
2. `$ echo Hello, World`
3. `$ echo Hello, world; Foo bar`
4. `$ echo Hello, world!`
5. `$ echo "line one";echo "line two"`
6. `$ echo "Hello, world > readme"`
7. `$ echo "Hello, world" > readme`
8. `$ cat readme`
9. `$ example="Hello, World"`
10. `$ echo $example`
11. `$ echo '$example'`
12. `$ echo "$example"`
13. `$ echo "Please enter your name."; read example`
14. `$ echo "Hello $example"`
15. `$ three=1+1+1;echo $three`
16. `$ bc`
17. `$ echo 1+1+1 | bc`
18. `$ let three=1+1+1;echo $three`
19. `$ echo date`
20. `$ cal`
21. `$ which cal`
22. `$ /bin/cal`
23. `$ $(which cal)`
24. `$ ` `` `which cal` ``

25. `$ echo "The date is $(date)"`
26. `$ seq 0 9`
27. `$ seq 0 9 | wc -l`
28. `$ seq 0 9 > sequence`
29. `$ wc -l < sequence`
30. `$ for I in $(seq 1 9) ; do echo $I ; done`
31. `$ (echo -n 0 ; for I in $(seq 1 9) ; do echo -n +$I ; done ; echo) | bc`

Hints

- Use the up/down arrows to browse the history of the shell to find recently used commands
- Use the left/right arrows (Pos1/End keys) to change the position of the text cursor
- Use the tab key to autocomplete commands when you're halfway through typing them - it saves time!
- The middle mouse key copies previously marked text into the shell
- Bash has certain special characters such as `;!<>` that can cause unexpected behaviour when included outside of quotes in a command.
- Use the command `$ help` and `$ man` to find out what a specific command does.

Submission (directory: `1/introductory-bash`)

`1/introductory-bash/fb.txt`	The feedback file for this exercise.
`1/introductory-bash/findings.txt`	An enumerated list that provides for each command a prose sentence what the commands did. Where your expectation differs from observed behavior, include a sentence describing what surprised you. (at least 200 words).

Further Reading

- Shell scripting (also available as PDF) `https://www.shellscript.sh/`
- Shebang Wikipedia: `https://en.wikipedia.org/wiki/Shebang_(Unix)`
- Commands: `https://maker.pro/linux/tutorial/basic-linux-commands-for-beginners`
- Command line structure: `https://www2.cs.duke.edu/csl/docs/unix_course/intro-14.html`
- `https://en.wikipedia.org/wiki/Filesystem_Hierarchy_Standard`

Task 2: Bash Research (9 Marks)

This is a more difficult **optional** task which can be done instead of Task 2

We would like for you to produce a short (about 1 page, at least 300 words) report on the core concepts behind bash and shown them in practice. Your report should give brief explanations and examples of the following concepts:

- Special characters and escaping
- Redirection
- Piping
- Variables

- Arithmetic
- Common Commands that you find useful `beyond our examples (ls, cd, mkdir, ...)`

You should include examples from the alternative (easy) task.
In your report you should consider circumstances in which easily avoidable errors could be encountered and subsequently avoided in your commands (for example, unescaped special characters such as `;!#&`)

Submission (directory: `1/introductory-bash`)

`1/introductory-bash/fb.txt`	The feedback file for this exercise.
`1/introductory-bash/report.txt`	The 300 words report describing the concepts relevant to Task 1.

Further Reading

- `https://learnxinyminutes.com/docs/bash/` - Usage examples of many common bash commands.
- `https://www.gnu.org/software/bash/manual/html_node/Quoting.html` - Nice guide to how quotes work in Bash
- `https://jvns.ca/blog/2017/03/26/bash-quirks/` - Article going over basics and some common quirks in the language which can trip up beginners.
- `https://www.geeksforgeeks.org/piping-in-unix-or-linux/` - Covers piping and its application.
- Advanced Bash-Scripting Guide: I/O Redirection: `https://www.tldp.org/LDP/abs/html/io-redirection.html`

Task 3: Setup of BigData Environments in a Virtual Machine (12 Marks)

To get started with a development environment, we provide a VirtualMachine that comes with the necessary software that was installed according to [1]. Also, familiarize with the typical remote connection ussing SSH [2].
You can import the imag either on your personal laptop or on the systems in the lab[1]. The image uses VirtualBox[2]. VirtualBox is a free virtualization environment allowing you to run virtual machines on your system (the host system).
The username and password is ubuntu.
Goal of this task is to familiarize with the image and briefly check the services that come with it.

Tasks

1. Familiarize with SSH [2]. Create a private/public key. (look for, e.g., PUTTY [3] in Windows)
2. Download one of the VirtualBox images `https://hps.vi4io.org/shared/cs3dp19.ova` (with command line interface) or `https://hps.vi4io.org/shared/programming.ova` with Gnome.
3. Import the VM into VirtualBox, run it. You may have to adjust the network settings (see [4.]).
4. Connect via SSH to the machine. Import the SSH key.
5. Test that Hadoop works in standalone mode:
   `$ hadoop fs -ls`

   This should show the files in the home directory. Then run:

   ```
   $ hadoop jar $HADOOP/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.2.1.jar
   grep  /input  /output 'allowed[.]*'
   ```
   You will then find output in the directory `/output`.

[1] In the labs store the data locally, though, as the network drive might have a performance issue
[2] `https://www.virtualbox.org/`

6. Test that Spark runs: `$ pyspark`. Then insert the following program:

```
rdd = sc.parallelize(["b", "a"])
sorted(rdd.map(lambda x: (x, 1)).collect())
```

   This should output: `[('a', 1), ('b', 1)]`

7. Test that MongoDB [6] works:

```
$ sudo service mongod start
$ ./mongo.py
$ sudo service mongod stop
```

Submission (directory: `1/bigdata`)

`1/bigdata/fb.txt`	The feedback file for this exercise.
`1/bigdata/software.txt`	Short comments regarding your steps and issues that arose.

Hints

- You can become root (the superuser) using the command `$ sudo` in the shell.
- The IP Address is shown with the command `$ ifconfig`.
- By default, the SSH port has been forwarded to the VirtualMachine, i.e., you have to use `$ ssh -p 2222 ubuntu@localhost`.

Further Reading

1. Installation for Hadoop in standalone mode `https://www.digitalocean.com/community/tutorials/how-to-install-hadoop-in-stand-alone-mode-on-ubuntu-18-04`
2. SSH Keygen `https://www.ssh.com/ssh/keygen/`
3. PuttyGen `https://www.ssh.com/ssh/putty/windows/puttygen`
4. VirtualBox Network `https://www.virtualbox.org/manual/ch06.html`
5. PySpark `https://spark.apache.org/docs/latest/api/python/index.html`
6. PyMongo `https://api.mongodb.com/python/current/`