Solution of Exercise Sheet 1

The exercise sheets of the operating systems course contain theoretical and practical exercises. For solving the practical exercises you need a UNIX shell. A very popular one is the Bash [1]. The Apple Mac OS X Terminal is sufficient for most practical exercises. The Windows command prompt and the Windows PowerShell are not sufficient for the exercises.

To prepare yourself, it is ideal, if you install on your system the Linux operating system. An installation in a virtual machine is sufficient. Easy to use distributions are, for example, Debian [2], Ubuntu [3], Manjaro [4] and Mint [5]. A free virtualization solution is VirtualBox [6]

Alternatively, you can work with a live system on CD, DVD, or USB flash memory drive. In this case no local installation is required. A powerful live distributions KNOPPIX [7].

[1] `http://tiswww.case.edu/php/chet/bash/bashtop.html`
[2] `http://www.debian.org`
[3] `http://www.ubuntu.com`
[4] `https://manjaro.org`
[5] `http://www.linuxmint.com`
[6] `http://www.virtualbox.org`
[7] `http://www.knopper.net/knoppix`

Exercise 1   (Batch Processing)

1. Describe the objective of batch processing.

   To maximize the CPU utilization.

2. Describe why batch processing causes an acceleration effect, when multiple tasks are executed.

   Because of the automation. No time is wasted when changing users.

3. Name the preconditions that must be satisfied for batch processing before the execution of a task can begin.

   Each program needs to be provided completely (with all input data!) before the execution may begin.

4. Name tasks for which batch processing is well suited.

   Batch processing is well suited for the execution of routine tasks.

5. Batch processing is always. . .

☐ interactive   ☒ non-interactive

6. Name an application of batch mode, which is still popular today.

   Batch files, shell scripts,...

7. Describe what spooling is.

   Spooling removes I/O workload from the CPU by using additional hardware.

Exercise 2   (Time-Sharing)

1. Describe the objective of time-sharing.

   A fair distribution of the computing time among the users.

2. Describe how time-sharing distributes the computing time among the processes.

   By using time slices.

3. Give the name of the quasi-parallel program or process execution.

   Multitasking.

4. Describe the objective of the quasi-parallel program or process execution.

   Minimizing the response time.

5. Describe what scheduling is.

   Automatic creation of an execution plan (schedule), which is used to allocate time limited resources to users or their processes.

6. Describe what swapping is.

   Process of storing and removing data to/from main memory from/into background memory (HDDs/SSDs).

7. Describe how memory protection works.

   The memory is split and running programs are separated from each other.

8. Describe the purpose of memory protection.

   This way, a bug or crash of a single program does not affect the stability of other programs and the total system.

Exercise 3   (Files and Directories)

1. Create in your home directory a directory `BTS`.

   `$ mkdir ~/BTS`

2. Navigate to the directory `BTS` and create inside an empty file with the filename `File1.txt`.

   - Do not use an editor application to create the file, but a command line command.

   `$ cd ~/BTS && touch File1.txt`

3. Check the file size of the file `File1.txt`.

   `$ ls -lh File1.txt`

4. Change the modification time of the file `File1.txt` to your birth date.

   `touch -t XXXXYYZZAABB File1.txt`

   XXXX *specifies the year.*
   YY *specifies the month.*
   ZZ *specifies the day of the month.*
   AA *specifies the hour.*
   BB *specifies the minute.*

5. Create a new file in the shell `File2.txt` and insert any text with more than just a single line as content into the new file.

   - Do not use an editor application to insert the text into the file, but a command line command.

   `$ echo -e "Line1\nLine2" > File2.txt`

6. Print out the first line of the file `File2.txt` in the shell.

   `$ head -n 1 File2.txt`

7. Append the content of `File2.txt` to `File1.txt`.

   - Do not use an editor application, but a command line command.

   `$ cat File2.txt >> File1.txt`

8. Create in your home directory, a directory with the directory name `BTS_new_semester`.

   `$ mkdir ~/BTS_new_semester`

9. Copy the files `File1.txt` and `File2.txt` from the directory `BTS` into the directory `BTS_new_semester`.

```
$ cp ~/BTS/* ~/BTS_new_semester
```

10. Erase the directory `BTS`.

```
$ rm -rf ~/BTS
```