

Assignment One

Designing and Developing a Process Scheduling Algorithm

Overview

A process scheduling algorithm deals with one fundamental question – which process in the ready queue should be allocated the CPU? The main goal of such an algorithm is to minimize resource starvation, and to ensure fairness amongst the processes vying for the CPU. In most cases, it is desirable to minimize the average turnaround, waiting, and response times. However, under certain circumstances, there may be a need to optimize the upper-bound values, e.g., the maximum waiting time, or maximum response time.

There have been many scheduling algorithms proposed, from the simple first-come, first-served and round-robin algorithms typically covered as textbook examples, to hybrid and advanced algorithms designed and developed by researchers and scientists worldwide. A cursory search for "*CPU scheduling algorithms*" on the IEEE Xplore Digital Library¹ returns more than 1,000 results on the topic. In this assignment, you and your team are to design and develop a process scheduling algorithm.

Task

In this assignment, you and your team are required to design and develop a process scheduling algorithm for CPU process scheduling. You are free to decide on your algorithm design, however for this assignment, the parameters to optimize are limited to the average and maximum turnaround time, and the average and maximum waiting time.

At the end of the assignment, you are required to submit 1) the source codes for your solution, and 2) a report detailing your solution in-depth. It is expected that your submission be of a high quality and demonstrates understanding of the topic at hand.

Assignment Deliverables and Deadline

You are required to submit, by the end of **Sunday, October 31, 2021**

- **Source codes** for your solution; and
- **A detailed report of between four (4) to six (6) pages** detailing your algorithm in-depth

More information regarding each item is described in the subsections to follow

¹ <https://ieeexplore.ieee.org/singaporetech/remotexs.co/>

Source codes

Your scheduling algorithm must be implemented in the **C programming language**². In order to be able to test your algorithm, you are to implement your algorithm as a Linux command-line application that takes in a filename as parameter³, and outputs the average and maximum turnaround time, and the average and maximum waiting time to the console.

Program restrictions

Your algorithm and command-line application must be implemented fully in the C programming language, and the instructor must be able to compile your code using only the GNU C Library⁴ on a Linux operating system. As the grading of your assignment will be semi-automated, you must adhere strictly to the requirements specified herein, else the grading system will not be able to pick up your program outputs⁵

- The entire source code for your algorithm and command-line application must be contained within a **single file** called `assignment.c`. Note that within your source code file, you are free to divide your algorithm and program into different functions.
- The grading system shall run the following command to compile your program

```
gcc -o assignment assignment.c
```

Should the compile fails, you will be given zero marks for the assignment.

- Your program must take in one, and only one command-line argument. This argument is a filename of an **input text file** containing a list of process arrival times, burst times, and priority levels. Input text files shall contain only ASCII characters (sometimes listed as ANSI encoding) and shall conform to the line feed (LF) end-of-line format.⁶
- The last line of the input text file shall be an empty line.
- Each line (except for the last line) in the input text file contains information for one process's arrival time, burst time, and priority level, in the format

```
<arrival_time><space><burst_time><space><priority_level>
```

² There are other, perhaps "easier" to use languages such as Python, Java, and C#. However as most operating systems are still written using languages such as C / C++, this assignment serves to uphold that spirit.

³ The file specified in the filename shall contain the list of processes and its arrival times, burst times, and priority levels.

⁴ The GNU C Library comes standard with pretty much most modern-day Linux distributions such as Ubuntu, Debian, etc.

⁵ Should the grading system fail to pick up the expected output lines, it will be assumed that your program failed to generate such a line, and consequently zero marks will be awarded for items associated with producing that line.

⁶ Sample input text files will be uploaded to the learning management system for reference.

Figure 1 shows an example of an input text file, which corresponding to Question 3 in the *Processes and CPU Scheduling, Part II* practise question set.

```
0 2 2
0 1 1
0 8 4
0 4 2
0 5 3
```

Figure 1

- For priority levels specified in the input text files, the lower the number, the higher the priority. A process with priority level 1 has a higher priority than a process with priority level 3.
- Your program must, at minimum, display the following 4 output lines, in exactly the following format shown

```
average turnaround time: <avg_turnaround_time>
maximum turnaround time: <max_turnaround_time>
average waiting time: <avg_waiting_time>
maximum waiting time: <max_waiting_time>
```

- As the system detects the presence of the phrases "average turnaround time", "maximum turnaround time", "average waiting time", and "maximum waiting time" in each output line to detect the presence of such values, do not use these phrases, whether in part or as a whole, in any other lines of your output except for the four output lines listed above.
- Figure 2 shows the expected input and (minimum) output of your application. The input text file in this example is named `input.txt`.

```
weihan@DESKTOP-7AVEQ08:~/assignment_one$ gcc -o assignment assignment.c
weihan@DESKTOP-7AVEQ08:~/assignment_one$ ./assignment input.txt
average turnaround time: 13.53
maximum turnaround time: 15.22
average waiting time: 8.10
maximum waiting time: 20.44
weihan@DESKTOP-7AVEQ08:~/assignment_one$
```

Figure 2

Report

Your report should utilize a template from a major scientific journals or proceedings, such as those from the IEEE^{7,8} or ACM⁹. All reports must be in **PDF** format.

⁷ https://www.ieee.org/conferences_events/conferences/publishing/templates.html

Proper citation and referencing must be maintained. You may use the citation format specified for the journal or proceedings template you are utilizing. Any form of academic misconduct or plagiarism **will be severely dealt with**.

Your report should detail your solution in-depth and should include, but not limited to

- A comprehensive background research of scheduling algorithms similar to your design¹⁰ with at least 8 references (credit will be given for appropriate use of sources from academic journals and / or conference proceedings);
- Analysis of the problem that you are trying to solve with your scheduling algorithm, i.e., why are you designing the algorithm, what are you trying to optimize;
- Your algorithm design, in an appropriate format, e.g., pseudocode, flowchart, etc. (you must provide the algorithm pseudocode in your report at minimum);
- Performance analysis of your algorithm, and comparison to other algorithms (include relevant Gantt charts, tables, and figures / graphs);
- Discussion of your algorithm's strengths and weaknesses.

A few sample conference papers will be uploaded to the learning management system for reference. Note that these conference papers do not reflect the best quality papers out there, and some of them contain severe readability issues and even algorithm errors. They are provided for you to learn from them, in terms of what is good to include and what pitfalls to avoid when writing your report.

Submission Information

Your source code and report must be uploaded to the designated Dropbox folder on the xSiTe Learning Management System, by the end of **Sunday, October 31, 2021**. The Dropbox folder shall be open for submission closer to the deadline.

Contact Information

Should you have any questions or queries regarding this assignment, please contact the instructor at Weihan.Goh@Singaporetech.edu.sg. Please note that the instructor may not be able to answer certain queries especially when they may give teams an unfair advantage over others.

⁸ https://www.ieee.org/publications_standards/publications/authors/author_templates.html

⁹ <https://www.acm.org/publications/proceedings-template>

¹⁰ For example, if you design a scheduling algorithm that is a variation of the shortest job first algorithm, you should provide background research on other, perhaps similar, algorithms based on the shortest job first algorithm.

Any academic misconduct or plagiarism will be severely dealt with.

END OF DOCUMENT