CPSC 457 S17 - Assignment 2

Due date: **Friday, June 2, 2017 at 11:59pm**.
Individual assignment. No group work allowed.
Weight: 8% of the final grade.

Link to the assignment web page:

https://sites.google.com/site/pfederlcpsc457spring2017/assignments/assignment-2

Q1 - Written question (4 marks)

Define DMA and multiprogramming. Would the concept of multiprogramming be practical on a computer that does not support DMA? Why or why not?

Q2 - Written question (3 marks)

Describe how a wrapped system call, e.g. `read()` in libc, invokes the actual system call in the kernel. Is it essential that a wrapper of a system call is named the same as the underlying system call?

Q3 - Written question (4 marks)

Is it possible for a process to go from the blocking state to the running state? What about going from the ready state to the blocking state? If yes, describe an example, if not, describe why not.

Q4 - Written question (3 marks)

What is a context switch? What are the actions taken, including the information to be saved, during a context switch?

Q5 - Programming question (10 marks)

Write a bash shell script "`scan.sh`" that takes two arguments:

1. suffix – a string
2. N – a positive integer

The shell script will recursively scan the current directory for all files ending with the given suffix, sort the files by their size, and print the the filenames of the largest N files to the standard output, followed by the sum of the sizes of the reported files. Each file reported should be followed by the size of the file in bytes. If the total number of files found is less then N, report all files. The listed files should be sorted by the size, in descending order.

Sample output:

```
$ scan.sh jpg 5
./face.jpg 88374
./a/b/school-jpg 88339
./a/tree.jpg 38883
./b/housejpg 3001
```

```
./a/b/jpg 233
Total size: 218830
```

Hints:

- You may consult the manual pages for one or more of the following utilities: `find, sort, head, awk`
- You may use pipes `"|"` to feed the output of one utility program into another utility program as input.
- You can assume there will be no spaces in any of the filenames or directory names.

Q6 - Programming question (15 marks)

Implement Q5 in C or C++, and name your program "`scan.c`" or "`scan.cpp`". Your program can make a one time use of the `system()` or `popen()` functions to invoke an external command to get the recursive list of files in the current directory. All other operations **must** be performed in C/C++, such as:

- determining file size,
- filtering filenames based on the extension,
- sorting,
- calculating the sum.

You may find it useful to use the following functions:

- `system()` to find all files recursively in a directory, and save them to a text file, e.g. `system("find . –type f > temp.txt");`
- `popen()` to find all files recursively, and read the results directly, e.g. `FILE * fp = popen("find . –type f", "r");`
- `stat()` to determine the size of the file;
- `qsort()` for C or `std::sort()` for C++, for sorting purposes.

You may assume that:

- the total number of files in the directory will be less than 1000, and
- the maximum pathname length of any file will be less than 512.

The output of your C/C++ program should be identical to the output of your bash script from Q5.

Q7 – Written question (3 marks)

Run `"strace -c"` and `"time"` on your bash script from Q5 and your C/C++ program from Q6 and compare the results. Explain why the results different? Include the output of the above commands in your report.

Q8 – Programming question (10 marks)

Write a multi-threaded solution for computing the sum of a large array of integers. Your program, "`sum.c`" or "`sum.cpp`", will take two arguments from the command line. The first argument is the name of the file containing the integers (one number per line) to be read into the

array, and the second argument, T, is the number of threads to be created. You may assume that the input contains at most 1,000,000 integers and the number of threads is always smaller than the number of integers in the file.

Your program will divide the array into T groups. All groups, except the last group, should contain the same number of integers. For example, given T=2 and the array [4,7,3,5,98,23,53], the first group contains (4,7,3,5) and the second group contains (98,23,53). Your program should display the sum of the integers assigned to each thread and finally the sum. The expected output for the above example is as follows, assuming the numbers are provided in `input.txt`:

```
$./sum input.txt 2
Thread 1: 19
Thread 2: 174
Sum = 193
```

The numbers after the word "Thread" in each line are the thread IDs. During the marking session your TAs will run your code on different input files and with different thread numbers, so it is important that your program accepts command-line arguments instead of hardcoded input file name and the number T.

Submission

You should submit 4 files for this assignment:

- Answers to the written questions combined into a single file, called either `report.txt` or `report.pdf`. Do not use any other file formats.
- Your solutions to Q5, Q6 and Q8 in a files called `scan.sh, scan.c, sum.c`.

Since D2L will be configured to accept only a single file, you will need to submit an archive, eg. `assignment2.tgz`. To create such an archive, you could use a command similar to this:

```
tar czvf assignment2.tgz report.pdf scan.sh scan.c sum.c
```

General information about all assignments:

- Due time: All assignments are due at 23:59 on the due date listed on the assignment. Late assignments or components of assignments will not be accepted for marking without approval for an extension beforehand. What you have submitted in D2L as of the due date is what will be marked.
- Extensions may be granted for reasonable cases, but only by the course instructor, and only with the receipt of the appropriate documentation (e.g. a doctor's note). Typical examples of reasonable cases for an extension include: illness or a death in the family. Cases where extensions will not be granted include situations that are typical of student life, such as having multiple due dates, work commitments, etc. Forgetting to hand in your assignment on time is not a valid reason for getting an extension.
- After you submit your work to D2L, make sure that you check the content of your submission. It's your responsibility to do this, so make sure that you submit your

assignment with enough time before it is due so that you can double-check your upload, and possibly re-upload the assignment.

- All assignments should include contact information, including full name, student ID and tutorial section, at the very top of each file submitted.
- Assignments must reflect individual work. Group work is not allowed in this class nor can you copy the work of others. For further information on plagiarism, cheating and other academic misconduct, check the information at this link: http://www.ucalgary.ca/pubs/calendar/current/k-5.html.
- You can and should submit many times before the due date. D2L will simply overwrite previous submissions with newer ones. It's better to submit incomplete work for a chance of getting partial marks, than not to submit anything.
- Only one file can be submitted per assignment. If you need to submit multiple files, you can put them into a single container. The container types supported will be ZIP and TAR. No other formats will be accepted.
- Assignments will be marked by your TAs. If you have questions about assignment marking, contact your TA first. If you still have questions after you have talked to your TA then you can contact your instructor.