CSC4005/7007 High Performance Computing (2012/13)

Assignment 1 - Sequential Searching

Specification

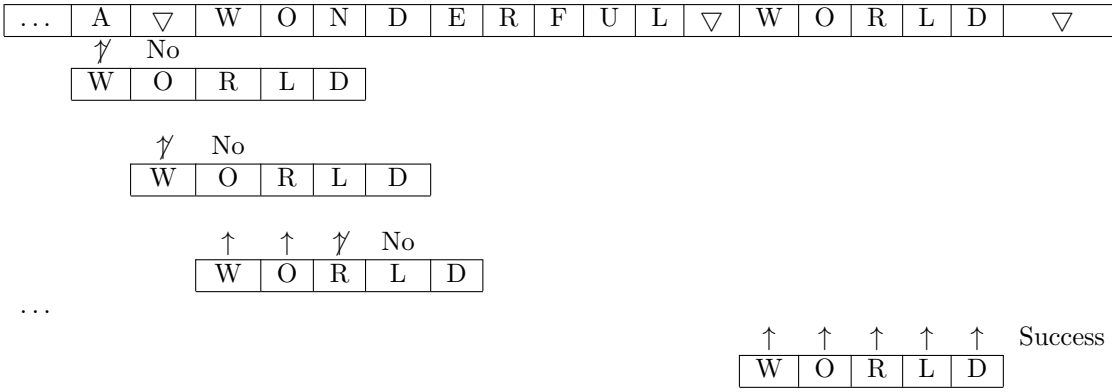
There are two non-empty sequences of characters known as the *text* and the *pattern*. The pattern is usually much smaller than the text. A program is required to determine whether the pattern occurs as a contiguous subsequence of the text and, if so, what is the position of the first character of the first occurrence.

Example

Suppose the text is `I THINK TO MYSELF WHAT A WONDERFUL WORLD`. If the pattern is `TERRIBLE` then it does not occur as a subsequence of the text. If the pattern is `WONDERFUL` then it occurs at position 25 (indexed from 0). It the pattern is `W` then it occurs at positions 18, 25 and 35 and the required answer is 18.

Algorithm

You are to use a *straightforward* pattern matching algorithm in which each potential position of the pattern is examined in turn



There are other, more sophisticated, algorithms with better worst case performance (e.g. *Boyer-Moore*), but in later assignments you will be asked to parallelise the straightforward algorithm rather than investigating other algorithms.

Implementation

You will be provided with a compressed tar file `Assignment1-dir.tar.gz`. This contains a program `searching_sequential.c` (which implements the straightforward algorithm) and some example test cases. The program looks for a folder called `inputs` in which is a collection of folders called `test0`, `test1`, `text2`, ..., one for each test case. Each test case folder contains the files `text.txt` and `pattern.txt` containing the test and pattern data. You may assume that all the characters are spaces or upper case letters. Two sample folders, `test0` and `test1`, have been provided for you, as has a compile script and a run script.

You should carry out each of the following steps:

- Transfer `Assignment1-dir.tar.gz` to `delllogin.qub.ac.uk` using WinSCP or another scp facility.
- Uncompress `Assignment1-dir.tar.gz` using `gunzip`, extract the contents of the archive file using `tar` and use `cd` to move into the `Assignment1-dir` directory.

```
$ gunzip Assignment1-dir.tar.gz
$ tar xvf Assignment1-dir.tar
$ cd Assignment1-dir
```

- Examine the contents of the directory and its subdirectories. In particular, examine the program `searching_sequential.c` and ensure that you understand how it works.
- Compile and run the program interactively, as follows,

```
$ ./compile searching_sequential
$ ./searching_sequential
```

- Run the program in batch using the `run.qsub` script. The output will be sent to the file `searching_sequential.out`. You can check on the status of the job using the command `qstat`. A job can be killed using `qdel jobid`. See `http://dellmaster1.qub.ac.uk/wiki/` for more info.

```
$ qsub  run.qsub
$ qstat -an -u compsc01
```

Assignment

- Determine 20 patterns and texts which will result in the worst case performance and create a scattergraph of CPU execution time plotted against length(text)*length(pattern). The product, length(text)*length(pattern), should be approximately equal to, 10^2, 10^4, 10^6, 10^8 and 10^{10}, and for each product you should construct 4 pattern-text examples. As a guide you should find that the worst case execution time for a product length of 10^{10} is over 10 seconds. The total file size should be no more than 100MBytes. Note: you will need to write a simple c program to generate the text and pattern files.
- Edit the compile script so that no compiler optimization is used i.e. use -O0 rather than -O2. Repeat your 20 tests.
- Repeat your 20 tests, this time using the gcc compiler with both -O0 and -O2.

Submission

You should submit a CD containing:

- Your test cases. Remember, there should be 20, and the total file size should be no more than 100MBytes.
- The source code and any scripts used.

You should also submit a short report in which you: present your 4 graphs; describe informally the characteristics of a worst case pattern and text; and draw any conclusions.