CS342 Operating Systems – Fall 2015
Project 1: Cooperating Processes

Assigned: Sep 23, 2015, Wednesday
Due date: Oct 03, 2015, Saturday, 11:55pm

Write a C program in Linux that will use multiple processes to search a set of input files for a given keyword. The program will be called **psearch** and it will take the following command line parameters: a keyword, number of input files (N), input filenames, an output filename. N will be >= 1. Each input file is an ascii text file (a sequence of lines). Each input file will be searched by a separate child process. The program (main process) will fork N children (child processes 0, 1, …, N-1). The main process will also create N temporary files to store the output of N child processes. Each child process will read its input file and search for the given keyword at each line and will emit a matching line together with a corresponding line number. When all processes finish, the main process will read the temporary files and will generate the output file. The output file will include information about where the given keyword appears in input files together with the corresponding lines. The format of the output file will be:

<inputfilename>: <line> (<linenumber>)

The output should be sorted according to first <inputfilename> and then <line>. The program will terminate after output is generated. An example invocation of your program can be:

psearch hello 2 in1 in2 out

The keyword is "hello". There are 2 input filenames, in1 and in2, and one output filename, out. The program will create 2 children. Assume file in1 contains:

```
hi hi
hi hello hi
hi hi hi
```

and file in2 contains:

```
hi
hi hello helloankara
alohellob
```

then the output file will contain:

```
in1: hi hello hi (2)
in2: alohellob (3)
in2: hi hello helloankara (2)
```

A filename can be at most 127 characters long. The maximum value of N can be 20. The number of lines in an input file can be anything. The maximum length of a line is 80 (excluding newline character).

Do also some *experiments*. For example, time the execution of your program for several cases. For example, you can time the execution of your program for different values of N and for different input sizes (number of lines in input file). You also run commands like ps, top, etc. and see the processes created and their resource usage. Put the results of your experiments into tables or charts which are to be put into a report. That means write a report about your experiments and results. Covert the report file to PDF before submission.

You will put your program C file (psearch.c) in a directory. In that directory you will also include a Makefile to compile your program, and a report.pdf file. Also include a README file with your Student ID in it. Pack (using tar command) and zip (using gzip command) your project folder containing the files mentioned and upload it as a single project1.tar.gz file in Moodle. Both your report and program (source code, execution, results) will be graded.