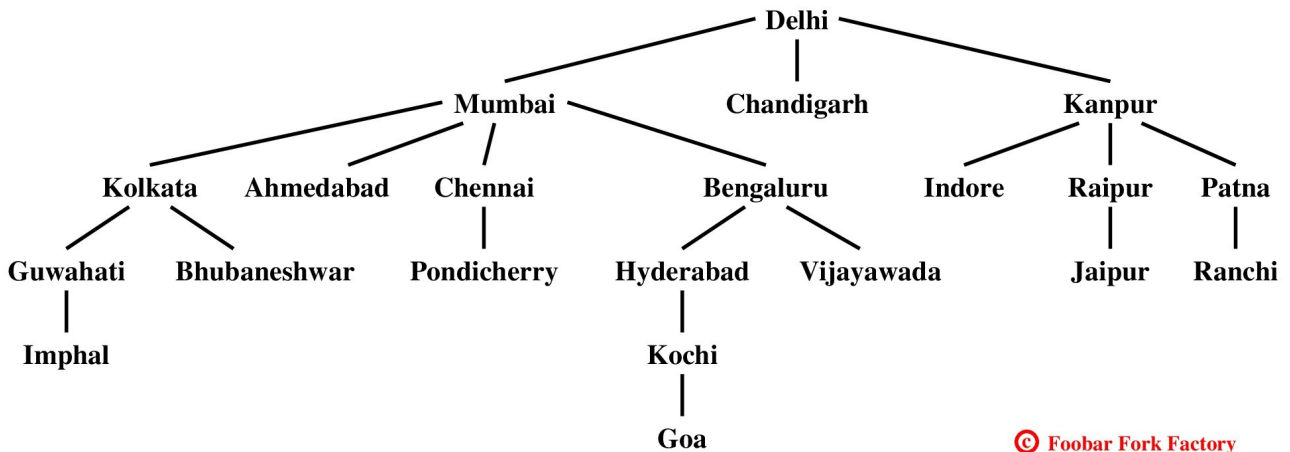


CS39002 Operating Systems Laboratory  
Spring 2024

Lab Assignment: 1  
10-Jan-2024

Introduction to multi-process applications

Foobar Fork Factory (FFF) is a famous manufacturer of silverware (cutlery in UK English). For logistic purposes, FFF has set up offices in many cities. These city offices are organized in the form of the following tree (produced here with permission from FFF).



© Foobar Fork Factory

This tree is stored in a text file *treeinfo.txt* in the following format. Each line of the file begins with the name of a city followed by its number of children, followed by the names of the children cities. Single spaces are used to separate the fields in each line. The city names may appear in any order. The alphabetic listing is given below. The child names of a city may appear in any arbitrary order (not alphabetic in the listing below).

The file *treeinfo.txt*

```
Ahmedabad 0
Bengaluru 2 Hyderabad Vijayawada
Bhubaneshwar 0
Chandigarh 0
Chennai 1 Pondicherry
Delhi 3 Mumbai Chandigarh Kanpur
Goa 0
Guwahati 1 Imphal
Hyderabad 1 Kochi
Imphal 0
Indore 0
Jaipur 0
Kanpur 3 Indore Raipur Patna
Kochi 1 Goa
Kolkata 2 Guwahati Bhubaneshwar
Mumbai 4 Kolkata Ahmedabad Chennai Bengaluru
Patna 1 Ranchi
Pondicherry 0
Raipur 1 Jaipur
Ranchi 0
Vijayawada 0
```

From time to time, some of the city offices need to update their financial reports. A city in the need of that prepares its updated report by contacting the cities recursively in the subtree rooted at that city. For example, suppose that the Mumbai office wants to update its report. It first contacts the Kolkata office. The Kolkata office first contacts the Guwahati office, and the Guwahati office contacts the Imphal office. Upon receiving the response from the Imphal office, the Guwahati office updates its report, and sends the updated report to the Kolkata office. Then, the Kolkata office contacts the Bhubaneshwar office, and upon the receipt of the response from Bhubaneshwar, prepares its report, and sends it to the Mumbai office. The Mumbai office then contacts the Ahmedabad office, and so on.

In this assignment, you write a multi-process application *proctree* to simulate the flow of work in the above report-update procedure in a city subtree (you do not prepare any reports though). For each city involved in the update procedure, a process is to be created. The process first opens the text file *treeinfo.txt*, and locates the line beginning with its city name. If no such line exists, the process terminates with an error message (see the sample output for the city name Kharagpur). If the line can be located, the process prints the name of the city and its PID at an appropriate indentation. Then, for each child (if any) of the city, it forks a new process, and waits until the child finishes. The child, in turn, exec's *proctree* with its city name. In order that the printing looks like a tree, each city process must know the level of the node in the subtree being printed. It is therefore necessary to send a second command-line argument to *proctree* for specifying this level (the first command-line argument is the city name). This second command-line argument can be optional and, if absent, is taken as 0 (meaning "no indentation").

In essence, this is nothing but a recursive pre-order printing of a subtree, that is, each node prints its name first, and then recursively prints the subtrees rooted at the children, in the sequence given in the input text file. However, instead of writing a recursive function, you perform the same task by creating a process tree. Each recursive call is replaced by a fork, and the child process initiates the recursive printing of the subtree rooted at that child, whereas the parent process waits until the child terminates (this is akin to waiting for a recursive call to return).

Write a C program *proctree.c*, and compile it to prepare an executable file *proctree*. Submit only the file *proctree.c*. Your program must use the `fork()`, `wait()`, and `exec()` calls in the manner specified in the last two paragraphs. You will be given no credit for solving the problem in any other manner.

**Note:** Use `atoi()` (declared in *stdlib.h*) for converting a numeric string to an integer. For converting an integer to a (numeric) string, use `sprintf()` (declared in *stdio.h*).

## Sample output

Each printed line starts with an indentation of four times the level of the node. The city name is then printed. Finally, the PID of the process is printed within parentheses. Each line is printed by one process. For example, in the example `./proctree Mumbai` below, the printing of the Kochi line is accomplished by the process `./proctree Kochi 3`. This process is forked by `./proctree Hyderabad 2`, has PID 11702, and the second argument 3 instructs the process to start printing after  $4 \times 3$  spaces.

```
$ ./proctree
Run with a node name
$ ./proctree Mumbai
Mumbai (11667)
    Kolkata (11668)
        Guwahati (11670)
            Imphal (11673)
                Bhubaneshwar (11676)
                    Ahmedabad (11680)
                        Chennai (11683)
                            Pondicherry (11687)
                                Bengaluru (11693)
                                    Hyderabad (11697)
                                        Kochi (11702)
                                            Goa (11706)
                                                Vijayawada (11711)
$ ./proctree Kanpur 2
Kanpur (11806)
    Indore (11807)
        Raipur (11808)
            Jaipur (11812)
                Patna (11817)
                    Ranchi (11823)
$ ./proctree Delhi
Delhi (11848)
    Mumbai (11849)
        Kolkata (11852)
            Guwahati (11864)
                Imphal (11885)
                    Bhubaneshwar (11891)
                        Ahmedabad (11896)
                            Chennai (11901)
                                Pondicherry (11910)
                                    Bengaluru (11915)
                                        Hyderabad (11920)
                                            Kochi (11924)
                                                Goa (11930)
                                                    Vijayawada (11933)
Chandigarh (11940)
Kanpur (11948)
    Indore (11951)
        Raipur (11955)
            Jaipur (11958)
                Patna (11964)
                    Ranchi (11968)
$ ./proctree Kharagpur
City Kharagpur not found
$
```