

CSI 402 – Program IV

Administrative Information:

- **Deadline:** 11 PM, Wednesday, April 26, 2006.
- Two parts, but just one makefile. (Additional specifications for makefile will be provided in the README file.)
- For part (a), one C source file is acceptable.
- For part (b), the client and server must both be split into two or more C source files.
- README file (by 10 PM, Apr. 13, 2006):

~csi402/public/prog4/prog4.README

Part (a): (Weightage: 25%)

- **Hidden files:** Files whose names start with ' . '.
- **Goal:** To print information about hidden files.

• **Command line:**

% p4a *pathname*

- Program must go through the files in the directory specified by *pathname*.
- For each hidden file, *except* "." and "..", program must print to stdout, the name of the file, the file size (in bytes) and the date of last modification (as specified in the handout).

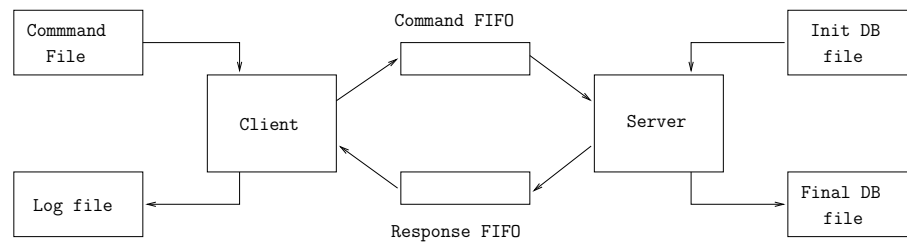
Suggestion: Study appropriate sections of the text by Haviland et al. (See handout for the list of sections.)

Errors to be detected: See handout.

Part (b): (Weightage: 75%)

Goal: To implement a simple client-server system using **named pipes (FIFOs)**.

Structure of the System:



Suggestion: Start by studying Chapters 5 and 7 of the text by Haviland et al. (Lecture notes and handouts for Lectures 14, 15 and 16 may also be useful.)

Notes:

- The client and server are separate programs.
- Your makefile must be such that when the grading script executes the command

```
% make p4b
```

two executable files, namely `p4b_server` and `p4b_client`, must be produced.

Command line:

```
% p4b_server initdb finaldb cmd log
```

Outlines for server and client:

– See page 6 of handout.

Example – Initial database file:

```
5
Hanks,Tom      2  311  403
Baker,Norma    3  311  402  403
Allen,Woody    1  404
Roberts,Julia  1  310
Pitt,Brad      0
6
310  3  TH--11:45-1:05
311  3  MW--12:30-1:25
402  3  TH--1:15-2:35
403  3  TH--10:15-11:35
404  3  MWF--1:40-2:35
445  3  T--4:15-7:15
```

Example – Command file:

```
addc  Hanks,Tom      404
addc  Martin,Steve   310
drpc  Baker,Norma    311
wdrw  Allen,Woody
wdrw  Allen,Tim
tcre  Roberts,Julia
newc  424  3    MWF--9:15-10:10
csch  310  MWF--1:25-2:20
ccre  402  4
gsch  445
gsch  426
gcre  446
gcre  402
```

Corresponding log file:

```
0  addc  1
1  addc  1
2  drpc  1
3  wdrw  1
4  wdrw  0
5  tcre  1    3
6  newc  1
```

```
7  csch  1
8  ccre  1
9  gsch  1    T--4:15-7:15
10 gsch  0    Error
11 gcre  0    -1
12 gcre  1    4
```

Final database file:

```
6
Hanks,Tom      3  311  403  404
Baker,Norma    2  402  403
Allen,Woody    0
Roberts,Julia  1  310
Pitt,Brad      0
Martin,Steve   1  310
7
310  3    MWF--1:25-2:20
311  3    MW--12:30-1:25
402  4    TH--1:15-2:35
403  3    TH--10:15-11:35
404  3    MWF--1:40-2:35
445  3    T--4:15-7:15
424  3    MWF--9:15-10:10
```

Assumptions for Part (b):

- Max. no. of students in the database = 100.
- Max. no. of courses in the database = 100.
- Max. no. of courses for which a student can be registered = 10.

Additional Notes:

- The command `addc` must fail if any of the following conditions hold.
 - (i) Adding the course will increase the number of courses in which the student is registered to 11.
 - (ii) Adding the course will increase the number of students in the database to 101.
- The command `newc` must fail if adding the course increases the number of courses in the database to 101.

- The `exit` command is *not* part of the command file.
- Use `stdio` library functions (e.g. `fopen`, `fscanf`) for database, command and log files. System calls (e.g. `open`, `read`, `write`) are needed for FIFOs.
- Define a suitable struct for sending a command to the server and another struct for receiving a reply from the server. Read/write FIFOs using these structs.

Errors to be detected: See handout.