Advanced Programming
COEN 11

Lab 7

Lab 7

- Restaurant Reservations
- Waiting List with File I/O
  - Extension of lab 5 or 6
    - Add saving/retrieving

Lab 7

- Initially
  - The waiting list may be either
    - empty
    - formed with information read from a file
- At the end
  - The updated waiting list is saved into a file

Lab 7

- The info should be saved in a text file according to the following format:

```
Name             Group Size
----------------------------------
Joe              5
Mary             3
Zoe              3
```

- It should be possible to read the file with commands such as cat and more

Lab 7

- The name of the file is an argument for the program
  - If the file does not exist
    - fopen returns NULL for reading
    - the list starts empty and is saved at the end into a file with the given name
  - If the file does exist
    - the list is initially formed with the information obtained from the file and is saved into the same file at the end

Lab 7

- The name of the file is an argument for the program
  - Example:

```
# ./wait_list <file_name>
```

or

```
# ./a.out <file_name>
```

Lab 7

- The name of the file is the first argument for the program
  - In the code:

```
main (int argc, char *argv[ ])
{
  . . .
  if (argc == 1)
  {
     printf (“The name of the file is missing!\n”);
     return 1;
  }

  read_file (argv[1]);
  …
```

Lab 7

- The name of the file is an argument for the program
  - In the code:
    - **argc** gives the number of arguments
    - **argv** is an array of strings, each of which is one of the arguments for the program
    - **argv[0]** is the name of the executable
    - **argv[1] – argv[argc – 1]** are the arguments

Lab 7

- The waiting list is created/modified interactively, as in lab 5 or 6, except that command quit (zero) will save the info into a file.
  - quit
    - save the list in the file specified and quit

Lab 7

- Requirements
  - Two new functions, called from main
    - Read from file
      - Receive file name as an argument
      - Call the insert function to insert the data read from file
    - Save to file
      - Receive file name as an argument

Lab 7

- Requirements
  - Use same insert function for inserting information from the file and from the keyboard.
  - Your insert function should have the following type:

    ```
    void insert (char *, int);
    ```
  - Read the name and number to local variables (char array and int) before calling the insert function.

Lab 7

- Requirements
  - Names cannot repeat!
    - Need to deal with that before calling function insert
  - Use function fseek to read the beginning of the file (header) before reading the data (names/numbers).
    - Type man fseek to learn how to use the function

Lab 7

- To receive full credit
  - Pre-lab (10%)
    - Test plan
  - Demo (30%)
    - Show the TA
      - Save to file
      - File is in the correct format
      - Retrieve from file
  - Submit to Camino (60%)

End

Lab 7