Advanced Programming

COEN 11, Fall 2014

Lab 7: File I/O and command line arguments

Restaurant Waiting List with file I/O and command line arguments

Business is blooming! Your restaurant is now very popular (every hipster in town must eat here!) and you managed to expand to a new bigger place. Update your reservation system as well by using files for your waiting list.
(Extension of project 5 or 6)

- Add saving/retrieving

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
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 unix commands such as cat and more

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

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

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

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;
}
else
     read_file (argv[1]);
…
```

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 (in our case only argv[1])**

The waiting list is created/modified interactively, as in project 5 or 6, except that command quit will save the info into a file.

- insert name number – insert a node with the name and number of people specified
- search size – extract (show and delete) oldest node with a number <= size
- list – print the list, name and number, from oldest to newest
- quit - save the list in the file specified and quit

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 (LIST*,char *, int);

Read the name and number to local variables (char array and int) before calling the insert function.

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