Advanced Programming
COEN 11

Lab 6

Lab 6

Waiting List by dept

- 1
- 2
- 3
- 4

Use an array of linked lists

- One list per dept

Lab 6

Same commands, adjust to the array of lists

- 1 name dept extra_info – insert a node in the dept list
  - index is dept – 1
- 2 – print the list for each dept: name, dept, and extra info
  - need to traverse all the lists
- 3 dept – show and delete the oldest node from the dept indicated
  - index is dept – 1
- 4 dept – print the entries corresponding to the dept indicated
  - index is dept – 1
- 5 name – show corresponding dept and info
  - need to traverse all the lists until the name is found
- 0 – save the info to the file, delete (free) all the nodes, and quit

Lab 6

Do not allow names to repeat

- Check all the lists before inserting

Keep your lists in the oldest-to-newest order

- Always insert a new entry at the end of the appropriate list
- Have tail pointers

To show the lists

- Traverse each list using a NODE pointer

To remove entries

- Traverse the appropriate list only.
- Change pointers to eliminate the node
- Free the node at the end

Lab 6

Requirements

- ➢Define a struct list
    - head and tail
- ➢Array of struct list, size = 4
    - heads and tails need to be initialized to NULL
- ➢Save the info and free all the nodes before quitting
    - New functions for option zero: save_file and delete_all

Lab 6

Extra Credit (10 points on the 1st midterm)

- ➢Add an option to change the dept, given a name
    - 6 name old_dept new_dept
        - Traverse the old_dept list searching for name
        - If found, the node moves to the end of the new_dept list

Lab 6

Saving the info to a file

- ➢Add saving and retrieving

Lab 6

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 6

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

```
Name    Dept    Info
-------------------------------------------
Joe     1       100.0
Mary    2       headache
Zoe     3       5
```

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

Lab 6

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 6

The name of the file is an argument for the program

- Example:

```
# ./wait_list  file_name
```

or

```
# ./a.out  file_name
```

Lab 6

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 6

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 6

The waiting list is created/modified interactively, except that command quit (zero) will save the info into a file.

- quit
  - save the list in the file specified, delete all the nodes, and quit

Lab 6

More Requirements

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

Lab 6

More 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, union info);
- Read the name and number to local variables (char array, int, and union) before calling the insert function.

Lab 6

More 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/extra info).
  - Use <man fseek> to learn how to use the function

Lab 6

To receive full credit

- Pre-lab (10%)
  - Test plan
- Demo (30%)
  - Show the TA
    - Start with an empty list
    - Add two people to depts 1, 2, and 3
    - Show each command
    - Quit
    - Start again
    - Show the list
- Submit to Camino (60%)