TRC2400 Computer Programming
ECE2071 Computer Organisation and Programming

Laboratory Session 5

Week 6 – Semester 1 2011

IMPORTANT – MARKING

You will receive marks for preliminary work and lab completion by completing quizzes on Blackboard
All quizzes receive equal marks and these will be scaled to give a final lab mark worth 10% of your final assessment.
You MUST complete the preliminary work quiz BEFORE midnight of the day before your lab otherwise you will receive a zero mark for the lab exercise (both preliminary and completion mark)
You must start the completion quiz before the end of your laboratory period (you will need the demonstrator to enter a password which will only be provided when you complete the lab)

1. Objectives

You are given several short programming exercises to complete. These are chosen so as to exercise and develop your understanding of

- dynamic memory allocation
- linked list data structure
- file input/output

2. Preliminary work

Before coming to the lab you should complete the preliminary work quiz on Blackboard. This week's quiz will cover your understanding of the bullet points above and also the exercises in this prac.

3. Programming Exercises

The overall aim of the lab is for you to add the functions output_list() and delete_node() to a C program that manages students information in a university. Student's information is kept in a text file, named "**student.dat**". Each line in the file contains student's name (first name, last name), ID number, course code (For example, 2320 for B.E, 0085 for Bsc/BE, 0333 for B.Tech) and telephone number. The maximum characters in a line are 80. For example, a line in the file "**student.dat**" contains the details in the format as:

```
Chris   Thomson  18344567  2320  95656777
Anna    Smith    19233350  0085  99055681
John    Kennedy  18345678  0333  97913467
```

The program must obtain the input filename from the user, read each line from the input file and construct a singly linked list where each node in the linked list will contain a student's first name, last name, ID number, course code and telephone number. The last node in the list should have a pointer field which points to NULL to indicate that it is the last node.

Each node in the linked list must have the following structure and user type definition:

```
struct record
        {
               char          *firstname;          /* first name */
               char          *lasttname;          /* last name */
              long int       idnumber;            /* ID number */
               int           ccode;        /* Course code */
               long int      phonenum;     /* telephone number */
               struct record *next;     /* pointer to structure */
```

```
            };
typedef struct record STUDENT;
```

Memory for the structure type STUDENT must be allocated at run time.

(Note: the *malloc()* function must be used to allocate nodes to the list and also to the character pointer fields to the first name and the last name in the structure).

The main program should call a function **make_node()** which takes the first name, last name, ID number, course code and the telephone number as its five arguments, allocates memory for the structure type STUDENT using malloc() function and returns a pointer to the newly allocated structure of type STUDENT.

After constructing the linked list data structure, the program should display the following menu on the screen:

MENU
1. Insert
2. Delete
3. Search
4. List
5. Save
6. Quit

Enter Your Choice:

If the user types 1, the program should execute **insert_node()** function, which should prompt the user to enter the new student's first name, last name, ID number, course code and the telephone number to be inserted at the start of the linked list.

If the user types 2, the program should execute the **delete_node()** function, which will prompt the user to select the student's last name so that the node corresponding to that student can be deleted from the linked list. The deleted node must be freed using **free()**.

If the user types 3, the program should execute the function **search_node()**, which will prompt the user to type the student's last name to be searched and then display his/her first name, last name, ID number, course code and the telephone number.

If the user types 4, the program should execute the function **print_list()**, which will display all student's first name, last name, ID number, course code and telephone number from the linked list on to the screen .

If the user types 5, the program should execute the function **output_list()**, which will write all student's first name, last name, ID number, course code and telephone number with the output format same as the input format on to the same text file, named "**student.data**".

If the user types 6, the program should exit without writing on to the file.

Your task for this lab is to complete the skeleton code provided on Blackboard.

4. Conclusion

Please note that marks will not be allocated to people who do not attend their allocated lab and complete the appropriate quizzes by their deadline. Under no circumstances will marks be recorded after the laboratory period is finished.

RAR 17/2/2009; WHL 24/03/2010; RAR 27/2/2011