

## Tutorial/Lab Session 3

### PURPOSE:

1. To practice C programming.
2. To learn how to write data to a file.
3. To learn how to read data from a file.

### PROCEDURE:

Practice 1: To create an application that has four program files.

Step 1: Login to the PC and start the X-window environment.

Step 2: Create a directory called lab3 (use “mkdir lab3”).

Step 3: Go to the directory lab3 (use “cd lab3”).

Step 4: Use “emacs” to create/edit the following four program files:

printname.c

```
#include <stdio.h>
#include "whoislecturer.h"

void PrintName()
{
    if (subject_code == 483)
        printf("\n M483 : Dr. Xie Ming");
    else
        printf("\n M432: Dr. Xie Ming");
    printf("\n value=%d, address=%ld",
        subject_code, &subject_code);
}
```

whoislecturer.c

```
#include <stdio.h>
#include "printname.h"

int subject_code ;

void main(int argc, char **argv()
{
    subject_code == 483 ;
    PrintName() ;
    subject_code = 432 ;
    PrintName() ;
}
```

printname.h

```
extern void PrintName() ;
```

whoislecturer.h

```
extern int subject_code ;
```

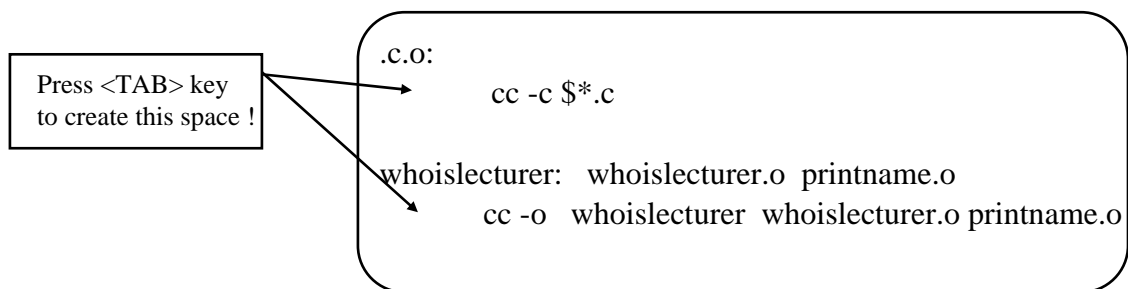
**NOTE:** Make sure that there is no typing error. Contents are case sensitive.

Practice 2: To create the “Makefile” for the previous application

Step 1: Move “mouse pointer” to a xterm and press the left button once. This makes the xterm active.

Step 2: Type “emacs Makefile” and press <ENTER> key.

Step 3: Key in the following content:



Step 4: Activate the menu “Files” and select the command “Save buffer” to save the above content to the file “Makefile”.

Step 5: Exit the “emacs”.

Step 6: Type “make” and press <ENTER> key.

Step 7: Type “whoislecturer” and press <ENTER> key. Observe what happens.

#### CREATIVE WORKS:

To modify the program files to display other messages.

Practice 3: To save data to a file.

Step 1: Move “mouse pointer” to a xterm and press the left button once. This makes the xterm active.

Step 2: Create and edit a file called “savedata.c” with the following content:

```
#include <stdio.h>
#include <string.h>
char data_string[100];
main(int argc, char **argv)
{
    FILE *pf;
    if (argc != 2)
    {
        printf("\n> Usage: <savedata your_data>\n");
        exit(0);
    }
    strcpy(data_string, argv[1]);
    pf = fopen("datafile.dat", "w");
    fwrite(data_string, sizeof(char), 100, pf);
    fclose(pf);
    printf("\n %s has been saved !\n", data_string);
}
```

Step 3: Edit a file called “Makefile” and add in the following content:

```
.c.o:
    cc -c *.c

savedata: savedata.o
    cc -o savedata savedata.o
```

Step 4: Exit “emacs”. Type “make savedata” and press <ENTER> key.

Step 5: Type “savedata abcdefgijkl” and press <ENTER> key.  
Observe what happens.

Step 6: Try to execute “savedata” with other strings (like, 12345678).

Practice 4: To read data from a file.

Step 1: Move “mouse pointer” to a xterm and press the left button once. This makes the xterm active.

Step 2: Create and edit a file called “readdata.c” with the following content:

```
#include <stdio.h>
#include <string.h>
char data_string[100];
main(int argc, char **argv)
{
    FILE *pf;
    pf = fopen("datafile.dat", "r");
    if (pf == NULL)
    {
        printf("\n> File: datafile.dat does not exist !\n");
        exit(0);
    }
    fread(data_string, sizeof(char), 100, pf);
    fclose(pf);
    printf("\n <%s> has been read in !\n", data_string);
}
```

Step 3: Edit a file called “Makefile” and add in the following content:

```
.c.o:
    cc -c *.c

readdata: readdata.o
    cc -o readdata readdata.o
```

Step 4: Exit “emacs”. Type “make readdata” and press <ENTER> key.

Step 5: Type “readdata” and press <ENTER> key.  
Observe what happens.

Step 6: Try to play with “savedata xxxx” and “readdata”.