

# Programming Project

Terry Marris January 2010

## 9 Menu

We integrate all the modules in the marathon menu program.

### 9.1 marathonenu.c

```
/* marathonmenu.c: controls marathon entries, times and results */
/* ver 1.1 24Jan2011 */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include "utility.h"
#include "entrybase.h"
#include "runnerbase.h"
#include "results.h"
#include "filename.h"

/* quit: terminates program execution */
void quit()
{
    exit(EXIT_SUCCESS);
}

/* SETUP */

/* setUp: sets file names */
int setUp()
{
    char *yesNo = malloc(5);
    FILENAME fName = readFromFile();
    printf("The current file names are:\n");
    printFileName(fName);
    yesNo = getStr("Change file names (y/n)?", 5);
    yesNo[0] = tolower(yesNo[0]);
    if (yesNo[0] == 'y') {
        fName = getFileNames(fName);
        writeToFile(fName);
        newEntryFile(entryFileName(fName));
        newRunnerFile(runnerFileName(fName));
    }
    return 0;
}
```

```

/* ENTRIES */

/* addMarathonEntry: adds new entries */
int addMarathonEntry()
{
    FILENAME fName = readFromFile();
    Entry entry = newEntry();
    entry = getEntry();
    addEntry(entry, entryFileName(fName));
    return 0;
}

/* withdrawEntry: marks an entry as withdrawn */
int withdrawEntry()
{
    FILENAME fName = readFromFile();
    char *number = getStr("Number?", NUMBERSIZE);
    amendEntryStatus(number, withdrawn, entryFileName(fName));
    return 0;
}

/* printMarathonEntry: prints all entries */
int printMarathonEntry()
{
    FILENAME fName = readFromFile();
    printEntryFile(entryFileName(fName));
    return 0;
}

/* TIMES */

/* startTime: sets race start time */
Time startTime()
{
    int hr, min, sec;
    Time time;
    FILE *file;

    hr = getInt("Start time: hour?");
    min = getInt("Minutes?");
    sec = getInt("Seconds?");
    time = setTime(hr, min, sec);
    file = fopen("starttime.dat", "wb");
    if (file == NULL) {
        printf("startTime: cannot open start time file");
        return defaultTime();
    }
    fwrite(&time, sizeof(Time), 1, file);
    fclose(file);
    return time;
}

```

```

/* finishTime: sets finish times */
int finishTime()
{
    int hr, min, sec;
    int number;
    Time start;
    Runner runner = newRunner();
    FILENAME fName = readFromFile();
    FILE *file;

    file = fopen("starttime.dat", "rb");
    if (file == NULL) {
        printf("finishTime: cannot open start time file");
        return 1;
    }
    fread(&start, sizeof(Time), 1, file);
    while (!feof(file))
        fread(&start, sizeof(Time), 1, file);
    fclose(file);
    printf("Enter 0 for runner number when finished\n");
    for ( ; ; ) {
        number = getInt("Number?");
        if (number < 1)
            break;
        hr = getInt("Finish hour?");
        min = getInt("Minutes?");
        sec = getInt("Seconds?");
        runner = setRunnerNumber(
            intToString(number), runner, entryFileName(fName));
        runner = setStartTime(start, runner);
        runner = setFinishTime(setTime(hr, min, sec), runner,
            entryFileName(fName));
        addRunner(runner, runnerFileName(fName), entryFileName(fName));
        printf("\n");
    }
    return 0;
}

/* printTimes: prints runners times */
int printTimes()
{
    FILENAME fName = readFromFile();
    printRunnerFile(runnerFileName(fName));
    return 0;
}

/* RESULTS */

/* halfMarathon: prints half marathon results */
int halfMarathon()
{
    FILENAME fName = readFromFile();
    makeResultsFiles(entryFileName(fName),
        runnerFileName(fName),
        halfMarathonFileName(fName),
        fullMarathonFileName(fName));
    printResultsFile(halfMarathonFileName(fName),
        "Half Marathon Results 2010");
    return 0;
}

```

```

/* fullMarathon: prints full marathon results */
int fullMarathon()
{
    FILENAME fName = readFromFile();
    makeResultsFiles(entryFileName(fName), runnerFileName(fName),
        halfMarathonFileName(fName), fullMarathonFileName(fName));
    printResultsFile(
        fullMarathonFileName(fName), "Full Marathon Results 2010");
    return 0;
}

/* MENUS */

/* entriesMenu: processes entries menu choice */
int entriesMenu()
{
    char *menu = "\nEntries: A(dd, W(ithdraw, P(rint, Q(uit: ";
    char *reply = getStr(menu, 5);

    reply[0] = tolower(reply[0]);
    switch (reply[0]) {
    case 'a':
        addMarathonEntry();
        break;
    case 'w':
        withdrawEntry();
        break;
    case 'p':
        printMarathonEntry();
        break;
    case 'q':
        return 0;
    default:
        break;
    }
    return 0;
}

/* timesMenu: processes times menu choices */
int timesMenu()
{
    char *menu = "\nTimes: S(tart, F(inish, P(rint, Q(uit: ";
    char *reply = getStr(menu, 5);
    reply[0] = tolower(reply[0]);
    switch (reply[0]) {
    case 's':
        startTime();
        break;
    case 'f':
        finishTime();
        break;
    case 'p':
        printTimes();
        break;
    case 'q':
        return 0;
    default:
        break;
    }
    return 0;
}

```

```

/* resultsMenu: processes results menu choices */
int resultsMenu()
{
    char *menu = "\nResults: H(alf marathon, F(ull marathon, Q(uit: ";
    char *reply = getStr(menu, 5);

    reply[0] = tolower(reply[0]);
    switch (reply[0]) {
    case 'h':
        halfMarathon();
        break;
    case 'f':
        fullMarathon();
        break;
    case 'q':
        return 0;
        break;
    default:
        break;
    }
    return 0;
}

/* mainMenu: processes main menu choices */
int mainMenu()
{
    char *menu =
        "\nMain: E(ntries, T(imes, R(esults, S(etup, Q(uit: ";
    char *reply = getStr(menu, 5);

    reply[0] = tolower(reply[0]);
    switch (reply[0]) {
    case 'e':
        entriesMenu();
        break;
    case 't':
        timesMenu();
        break;
    case 'r':
        resultsMenu();
        break;
    case 's':
        setUp();
        break;
    case 'q':
        quit();
        break;
    default:
        break;
    }
    return 0;
}

int main()
{
    for ( ; ; )
        mainMenu();
    return 0;
}

```

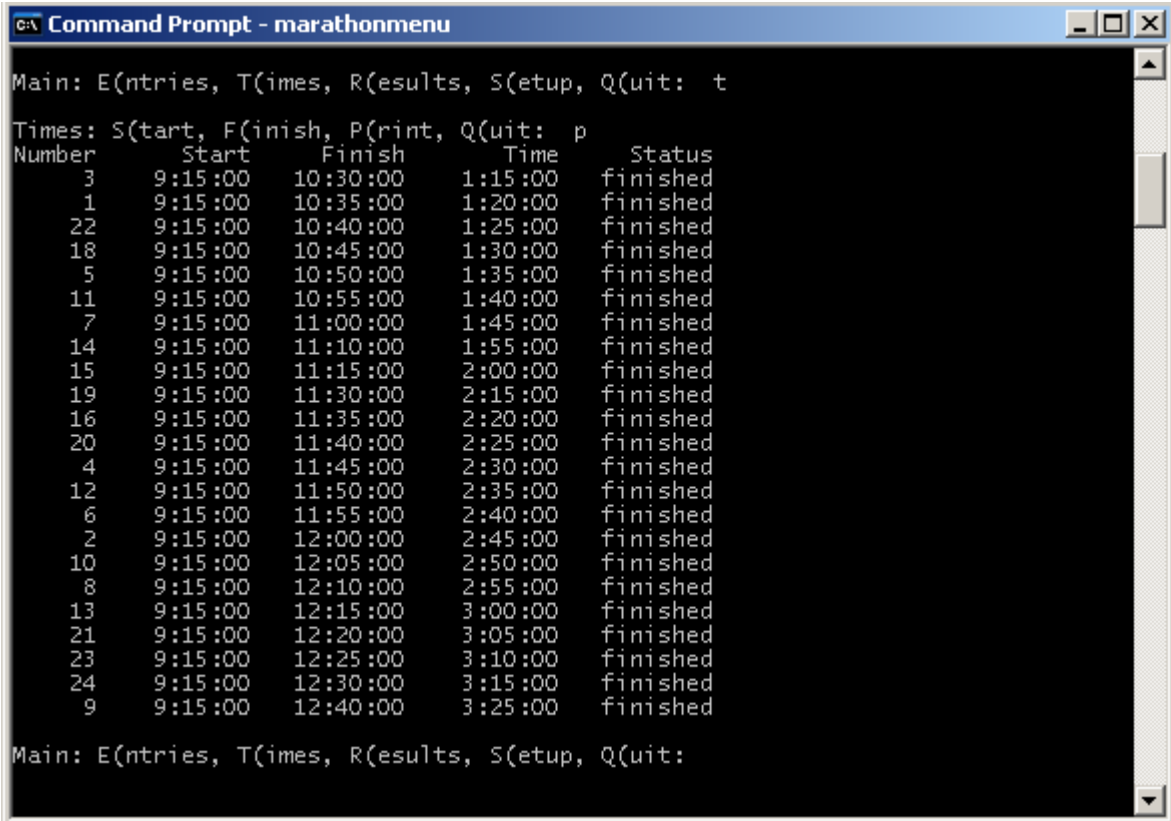
Each menu is shown in one line. The capital letters in each menu identify the options. Shown below is the main menu with options for entries, times, results, setup and quit. The entry option is selected and the entry menu is shown. Then, the print option is selected and the contents of the entry file is printed. Quit on the entries menu is selected and control returns to the main menu.

```

C:\ Command Prompt - marathonmenu
$ marathonmenu
Main: E(ntries, T(imes, R(esults, S(etup, Q(uit: e
Entries: A(dd, W(ithdraw, P(rint, Q(uit: p
Number Name Address Category Status
1 Justin Case 5 Kings Court half withdrawn
2 Barb Dwyer 3 Princes Parade full current
3 Stan Still 7 Princess Avenue half current
4 Terry Bull 2 Dukes Drive full current
5 Paige Turner 4 Queens Crescent half current
6 Mary Christmas 6 Princes Parade full current
7 Anna Sasin 6 Kings Parade half current
8 Doug Hole 8 Princes Place full current
9 Hazel Nutt 1 Princess Street half current
10 Stan Still 16 Dukes Road full current
11 Hazel Picking 11 Queens Road half current
12 Rose Bush 46 Ocean Boulevard full current
13 Hazel Bush 46 Ocean Boulevard full current
14 Pearl Button 12 Sea View half current
15 Jo King 25 Boulevard Crescent full current
16 Barry Cade 13 Harbour Wall half current
17 Mary Lee 15 Sea View full current
18 Carrie Oakey 14 Kings Court half current
19 Priti Manek 12 Princess Avenue full current
20 Tim Burr 1 Wood Lane half current
21 May Day 13 Sea View full current
22 Max Power 1 Boulevard Crescent half current
23 Rob Me 11 Wood Lane full current
24 Sue Me 11 Wood Lane full current
25 Andy Mann 19 Kings Court half withdrawn
Main: E(ntries, T(imes, R(esults, S(etup, Q(uit:

```

The times option is selected, and, in the times menu, the print option is chosen.



```
c:\ Command Prompt - marathonmenu
Main: E(ntries, T(imes, R(esults, S(etup, Q(uit): t
Times: S tart, F(inish, P(rint, Q(uit): p
Number   Start      Finish      Time       Status
3        9:15:00    10:30:00    1:15:00    finished
1        9:15:00    10:35:00    1:20:00    finished
22       9:15:00    10:40:00    1:25:00    finished
18       9:15:00    10:45:00    1:30:00    finished
5        9:15:00    10:50:00    1:35:00    finished
11       9:15:00    10:55:00    1:40:00    finished
7        9:15:00    11:00:00    1:45:00    finished
14       9:15:00    11:10:00    1:55:00    finished
15       9:15:00    11:15:00    2:00:00    finished
19       9:15:00    11:30:00    2:15:00    finished
16       9:15:00    11:35:00    2:20:00    finished
20       9:15:00    11:40:00    2:25:00    finished
4        9:15:00    11:45:00    2:30:00    finished
12       9:15:00    11:50:00    2:35:00    finished
6        9:15:00    11:55:00    2:40:00    finished
2        9:15:00    12:00:00    2:45:00    finished
10       9:15:00    12:05:00    2:50:00    finished
8        9:15:00    12:10:00    2:55:00    finished
13       9:15:00    12:15:00    3:00:00    finished
21       9:15:00    12:20:00    3:05:00    finished
23       9:15:00    12:25:00    3:10:00    finished
24       9:15:00    12:30:00    3:15:00    finished
9        9:15:00    12:40:00    3:25:00    finished
Main: E(ntries, T(imes, R(esults, S(etup, Q(uit):
```

Results is selected from the main menu. Then, half marathon is selected from the results menu.

```

C:\ Command Prompt
Main: E(ntries, T(imes, R(esults, S(etup, Q(uit: r
Results: H(alf marathon, F(ull marathon, Q(uit: h
Half Marathon Results 2010
Number Name          Time
  3 Stan Still        1:15:00
  1 Justin Case       1:20:00
 22 Max Power         1:25:00
 18 Carrie Oakey      1:30:00
  5 Paige Turner      1:35:00
 11 Hazel Picking     1:40:00
  7 Anna Sasin        1:45:00
 14 Pearl Button      1:55:00
 16 Barry Cade        2:20:00
 20 Tim Burr          2:25:00
  9 Hazel Nutt        3:25:00

Main: E(ntries, T(imes, R(esults, S(etup, Q(uit: r
Results: H(alf marathon, F(ull marathon, Q(uit: f
Full Marathon Results 2010
Number Name          Time
 15 Jo King           2:00:00
 19 Priti Manek       2:15:00
  4 Terry Bull        2:30:00
 12 Rose Bush         2:35:00
  6 Mary Christmas   2:40:00
  2 Barb Dwyer        2:45:00
 10 Stan Still        2:50:00
  8 Doug Hole         2:55:00
 13 Hazel Bush        3:00:00
 21 May Day           3:05:00
 23 Rob Me            3:10:00
 24 Sue Me            3:15:00

Main: E(ntries, T(imes, R(esults, S(etup, Q(uit: q
$

```

Finally, quit is selected and control returns to the command line.

We should really systematically test each option, looking for errors, here, but ...



## 9.2 make.bat

Typing in the commands for compiling and linking the modules quickly became non-trivial as the modules were developed. So a batch file, named *make.bat* and shown below, was created to help automate the compiling and linking process.

```
ECHO OFF
gcc -c utility.c -ansi -Wall -pedantic -o utility.o
gcc testutilty.c -ansi -Wall -pedantic utility.o -o testutility.exe
gcc -c entry.c -ansi -Wall -pedantic -o entry.o
gcc testentry.c -ansi -Wall -pedantic utility.o entry.o -o testentry.exe
gcc -c entrybase.c -ansi -Wall -pedantic -o entrybase.o
gcc testentrybase.c -ansi -Wall -pedantic utility.o entry.o entrybase.o -o
testentrybase.exe
gcc -c runner.c -ansi -Wall -pedantic -o runner.o
gcc testrunner.c -ansi -Wall -pedantic utility.o entry.o entrybase.o
runner.o -o testrunner.exe
gcc -c runnerbase.c -ansi -Wall -pedantic -o runnerbase.o
gcc testrunnerbase.c -ansi -Wall -pedantic utility.o entry.o entrybase.o
runner.o runnerbase.o -o testrunnerbase.exe
gcc -c result.c -ansi -Wall -pedantic -o result.o
gcc -c results.c -ansi -Wall -pedantic -o results.o
gcc testresults.c -ansi -Wall -pedantic utility.o entry.o entrybase.o
runner.o runnerbase.o result.o results.o -o testresults.exe
gcc -c filename.c -ansi -Wall -pedantic -o filename.o
gcc testfilename.c -ansi -Wall -pedantic utility.o filename.o -o
testfilename.exe
gcc marathonmenu.c -ansi -Wall -pedantic utility.o entry.o entrybase.o
runner.o runnerbase.o result.o results.o filename.o -o marathonmenu.exe
ECHO ON
```