Stop Watch / Clock

NAME: G. Shanmukha Reddy

ORGANIZATION: Micro IT

TOPIC: Stop Watch / Clock

Introduction

•This application combines two tools:
➤ A **clock** that shows the current time
➤ A **stopwatch** that tracks how long something takes
•It works directly from the **command line (terminal)**

What It Can Do

•Shows **current system time** in **12-hour format** with AM/PM

•Lets the user:

- Start the stopwatch
- Pause it
- Reset it
- Quit the program

•Displays **hours, minutes, and seconds**

How It Works

•Uses the C language and:

•Time. h for time tracking
•Conio. h for real-time key input
•Windows. h for screen clearing and delay

•Continuously updates the screen with the clock and stopwatch

Sample Output

Current Time: 03:25:42 PM
Stopwatch: 00:00:08
Press 's' to Start, 'p' to Pause, 'r' to Reset, 'q' to Quit

Why This Project is Useful

- Helps measure time during:
  - Workouts
  - Study sessions
  - Cooking
  - Experiments
- Practical and improves time-handling skills in coding

What I Learned

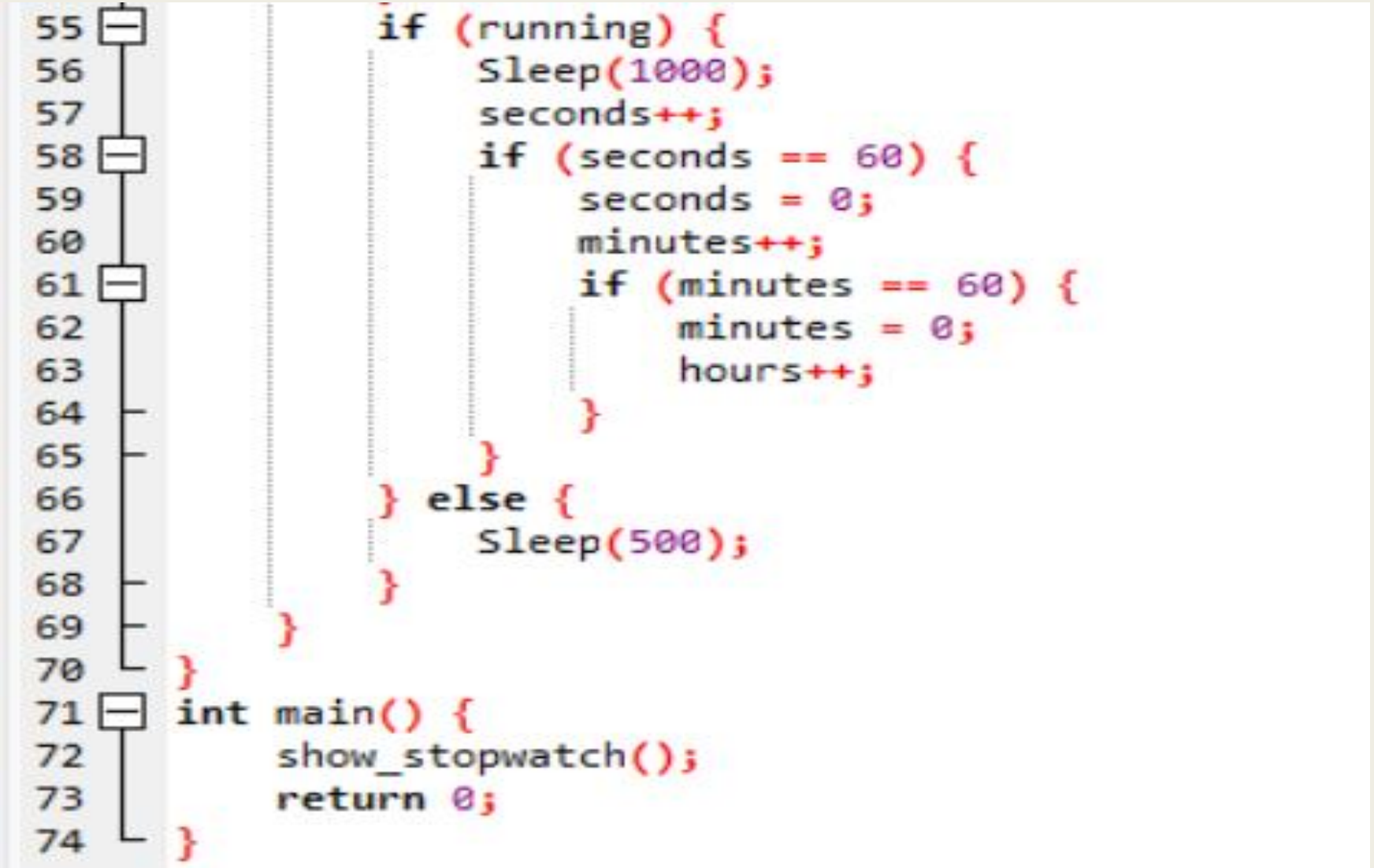
- How to handle **real-time updates**
- How to read and format **system time**
- Using user input without Enter key
- Building interactive console applications

Conclusion

- A simple yet powerful tool
- Easy to use and helpful in daily tasks
- Good practice for **C programming**, **logic building**, and **real-time control**

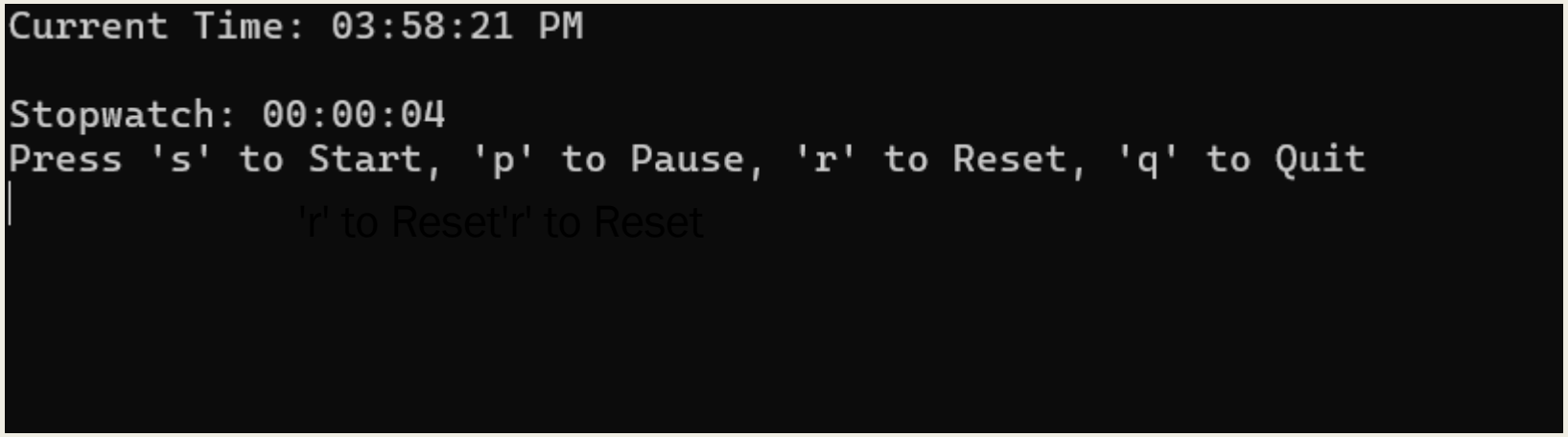
```c
#include <stdio.h>
#include <time.h>
#include <conio.h>
#include <windows.h>
void show_clock() {
    time_t rawtime;
    struct tm * timeinfo;
    char *ampm;


    time(&rawtime);
    timeinfo = localtime(&rawtime);

    int hour12 = timeinfo->tm_hour;
    if (hour12 >= 12) {
        ampm = "PM";
        if (hour12 > 12)
            hour12 -= 12;
    } else {
        ampm = "AM";
        if (hour12 == 0)
            hour12 = 12;
    }

    printf("Current Time: %02d:%02d:%02d %s\n",
           hour12, timeinfo->tm_min, timeinfo->tm_sec, ampm);
}
```

```c

void show_stopwatch() {
    int hours = 0, minutes = 0, seconds = 0;
    int running = 0;
    char ch;

    printf("\nStopwatch Controls:\n");
    printf("s = Start, p = Pause, r = Reset, q = Quit\n");

    while (1) {
        system("cls");
        show_clock();
        printf("\nStopwatch: %02d:%02d:%02d\n", hours, minutes, seconds);
        printf("Press 's' to Start, 'p' to Pause, 'r' to Reset, 'q' to Quit\n");
        if (_kbhit()) {
            ch = getch();
            if (ch == 's') {
                running = 1;
            } else if (ch == 'p') {
                running = 0;
            } else if (ch == 'r') {
                hours = minutes = seconds = 0;
                running = 0;
            } else if (ch == 'q') {
                break;
            }
        }
```

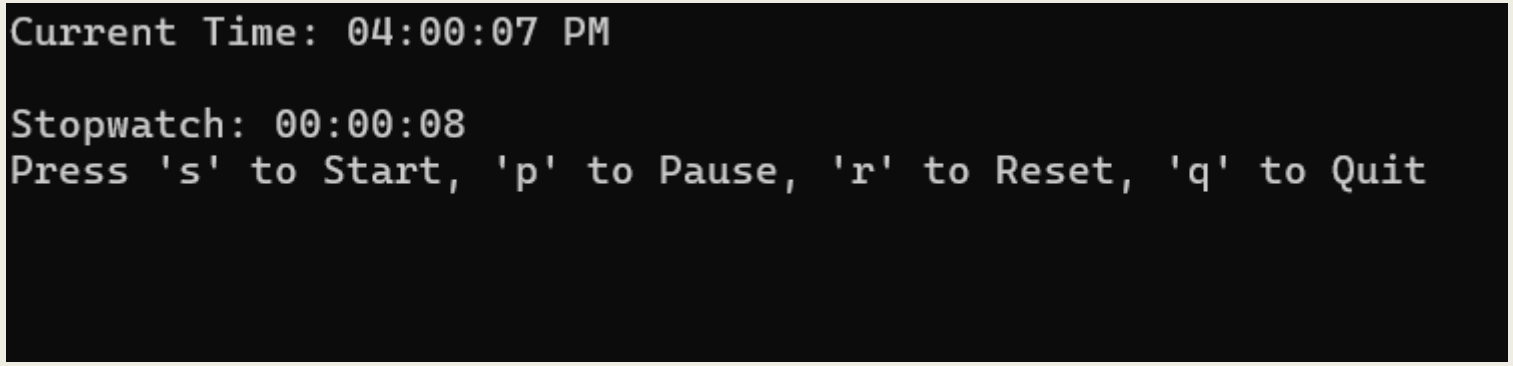


```
55          if (running) {
56              Sleep(1000);
57              seconds++;
58              if (seconds == 60) {
59                  seconds = 0;
60                  minutes++;
61                  if (minutes == 60) {
62                      minutes = 0;
63                      hours++;
64                  }
65              }
66          } else {
67              Sleep(500);
68          }
69      }
70  }
71  int main() {
72      show_stopwatch();
73      return 0;
74  }
```

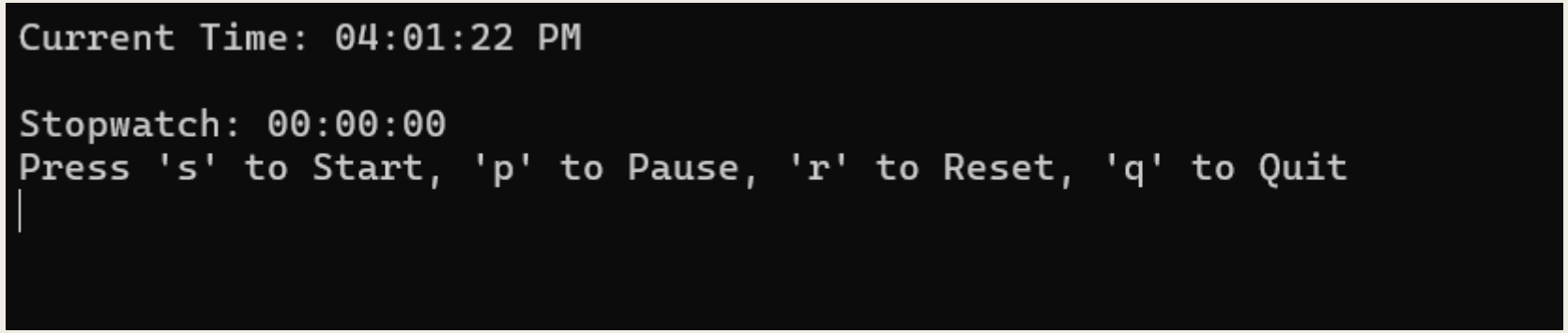
Press 's' to Start



'p' to Pause,



'r' to Reset



Thank You