FINAL REPORT OF COMPUTER ORGANIZATION AND ASSEMBLY LANGUAGE

PROJECT NAME:

SIMPLE CLOCK

Submitted by:

Syed Muhamamd Azhar Ali Naqvi

Id: csc-18f-053

Simple Clock

Abstract

There are primarily two types of clocks. One being analog and the other being digital. The analog clock was the only type of clock in the past when there were no digital technology available. But with the dawn of digital era, digital clocks have become one of our daily necessities. We use digital clock in different forms like hand watch, cell phones, personal computers etc.

We see digital clocks in our personal computers everyday. It is a prime component of these devices. In this project we have implemented a digital clock using the graphics mode of assembly language programming. Here we show how this can be done and walk you through the whole process.

Introduction

Assembly language is one the earliest means of computer programming. It is the lowest level of programming above machine language programming. Assembly programming requires deep understanding about the hardware we are dealing with. The hardware components can be directly manipulated using this. Since the programmer can play with the components as he wishes, using this programming method can get us rid of a lot of redundancy and thus create faster programs.

A simple clock is a type of clock that displays the Current time, as opposed to an analog clock, where the time is indicated by the positions of rotating hands.

To represent the time. It shows hours, minutes and seconds and can be set to show in both 24 hours format.

Features

The main and significant features of this program is described below.

1. The clock shows the time of the computer system.
2. It shows hours, minutes and seconds.
3. It uses 24 hour format.
4. It is developed in VGA graphics with 320x200 resolution.
5. It is displayed in a classy, stylish and attractive font.
6. The interface is simplistic with white text on black background.

Key concept:

Interrupt:

In systems programming, an interrupt is a signal to the processor emitted by hardware or software indicating an event that needs immediate attention. An interrupt alerts the processor to a high-priority condition requiring the interruption of the current code the processor is executing. The processor responds by suspending its current activities, saving its state, and executing a function called an interrupt handler (or an interrupt service routine, ISR) to deal with the event. This interruption is temporary, and, after the interrupt handler finishes, the processor resumes normal activities.

PROJECT CODE:

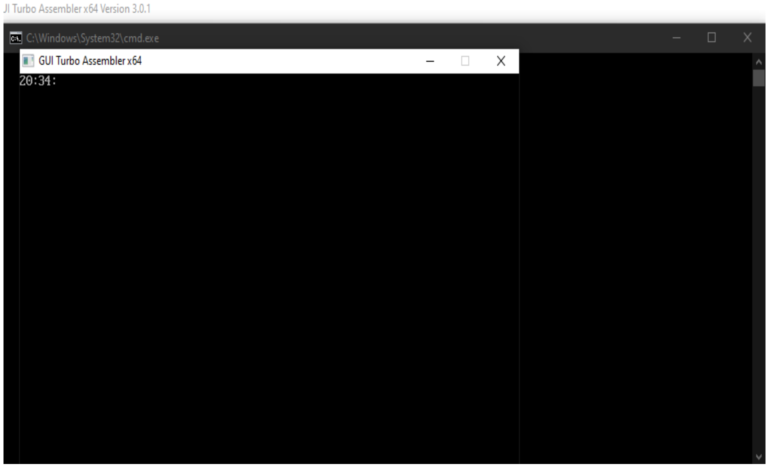
SOUCE CODE:

```
.model small
.stack 100h
.DATA
```

```
.CODE

START:
MOV AX,@DATA
MOV DS,AX

L1:
MOV AH,2CH
INT 21H
MOV AL,CH
CALL CONV
CALL DISP
MOV DL,':'
MOV AH,06H
INT 21H
MOV AL,CL
CALL CONV
CALL DISP
MOV DL,':'
MOV AH,06H
INT 21H
MOV DL,0DH
MOV AH,06H
INT 21H
JMP L1
MOV AH,4CH
INT 21H
```

```
DISP PROC NEAR
PUBLIC DISP
MOV BL,AH
MOV DL,AL
ADD DL,30H
MOV AH,06H
INT 21H
MOV DL,BL
ADD DL,30H
MOV AH,06H
INT 21H
RET
DISP ENDP
CONV PROC NEAR
 PUBLIC CONV
MOV AH,00H
MOV BH,0AH
DIV BH
RET
CONV ENDP

END START
```

OUTPUT:


JI Turbo Assembler x64 Version 3.0.1
C:\Windows\System32\cmd.exe
GUI Turbo Assembler x64

```
20:34:
```