**OPERATING SYSTEM**

Mini Project on

CREATING A Mini-OS

Submitted by :

MANISH PAWAR

RAHUL PANGA

DEVESH PATIL

Date: Sign:

**AIM**: Basically to create a mini-OS by creating a kernel code,a virtual floppy drive,and then building them together to output a simple bootloading screen which loads OS.

**OBJECTIVES:**

* This can be used to create operating systems acc. to our needs.
* This provides following properties to operator:

- Flexible kernel viz. free to code kernel

-Create a virtual drive for data on system only

-Create drive of any size to fit OS and its data

-Build the system

-Boot the floppy drive and run OS

It can be modified further for file & drive systems,authentication,etc.

**SOFTWARE AND HARDWARE REQUIREMENTS:**

**SOFTWARE REQUIREMENTS :**

● Operating system : Windows 10/R/8

● Development tools: Bochs 2.6,OS development kit,Floppy img ,virtual drive-vdfwin

● Programming language and libraries : C for kernel,Batch scripts for Build file

**HARDWARE REQUIREMENTS :**

● Operating System : Windows 10 pro 64 bytes.

● Processor : Intel core i5-2520M 2.5GHz processor.

● Storage : according to need

● Ram : 1024MB RAM Memory.

● Graphics : Intel HD Graphics 3000

**THEORY**

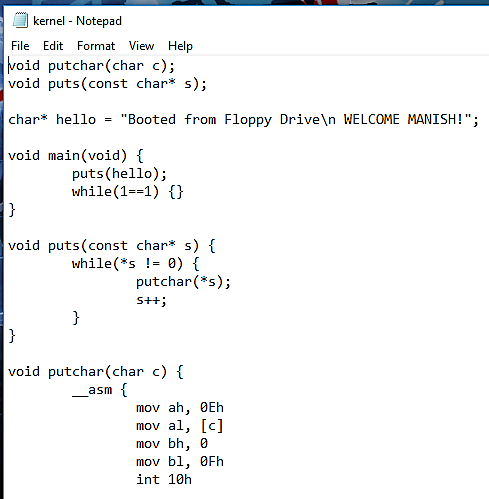
**Proposed system:**

In proposed system,we code for kernel in basic C along with some assembly instructions and then create a virtual floppy drive using vfdwin.exe application.Then we debug our code and boot floppy and use Bochs 2.6.8 to run our OS and thus display a message on bootstrap screen.Its much convenient than hard & scratch coding.

Proposed system provides with following solutions:

1. Easy and convenient to workout
2. Flexible to modify kernel
3. No need of physical drive and size can be suited to our demands
4. Easy to debug and run as all steps are available in clicks
5. Pre-requisite is only C and some scripts

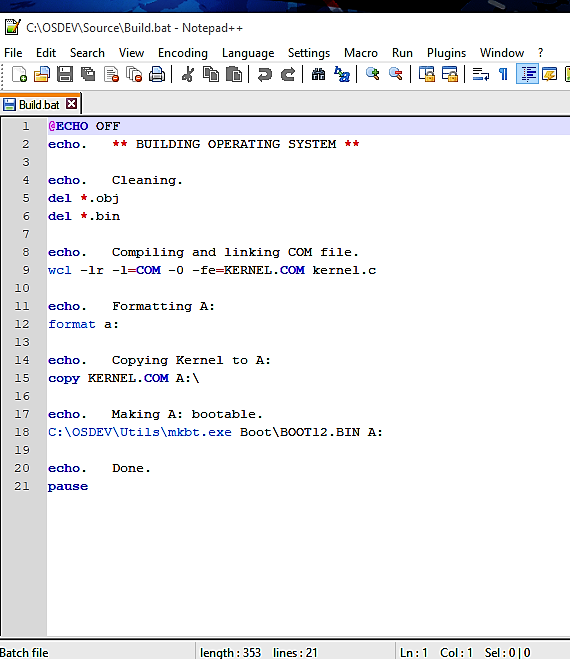
**SYSTEM DESIGN**

First of all, install OS development kit & then go to its directory and perform steps in that directory.

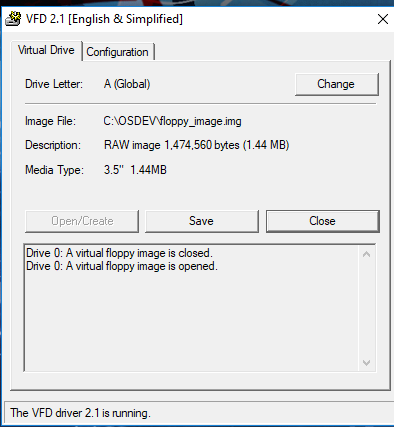
Write a C kernel

code and save it

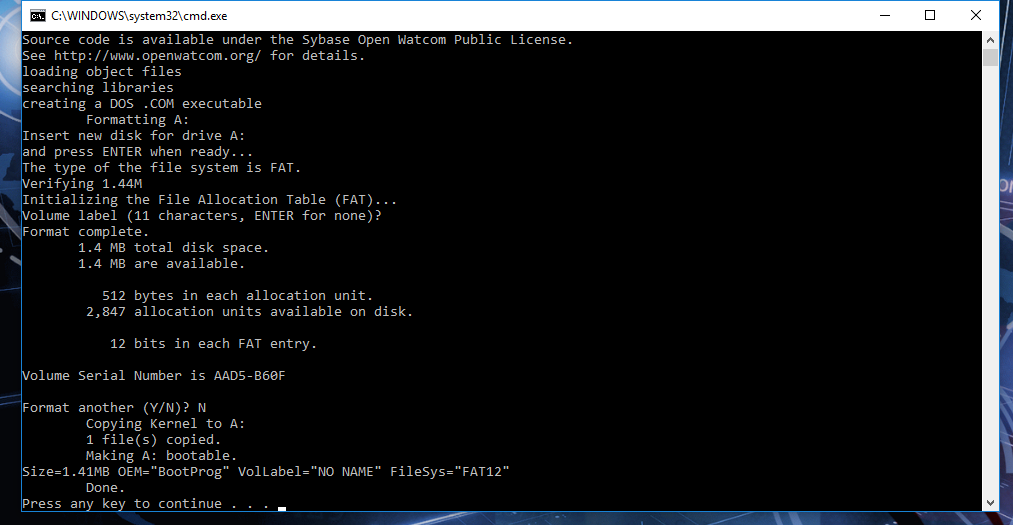
as .c



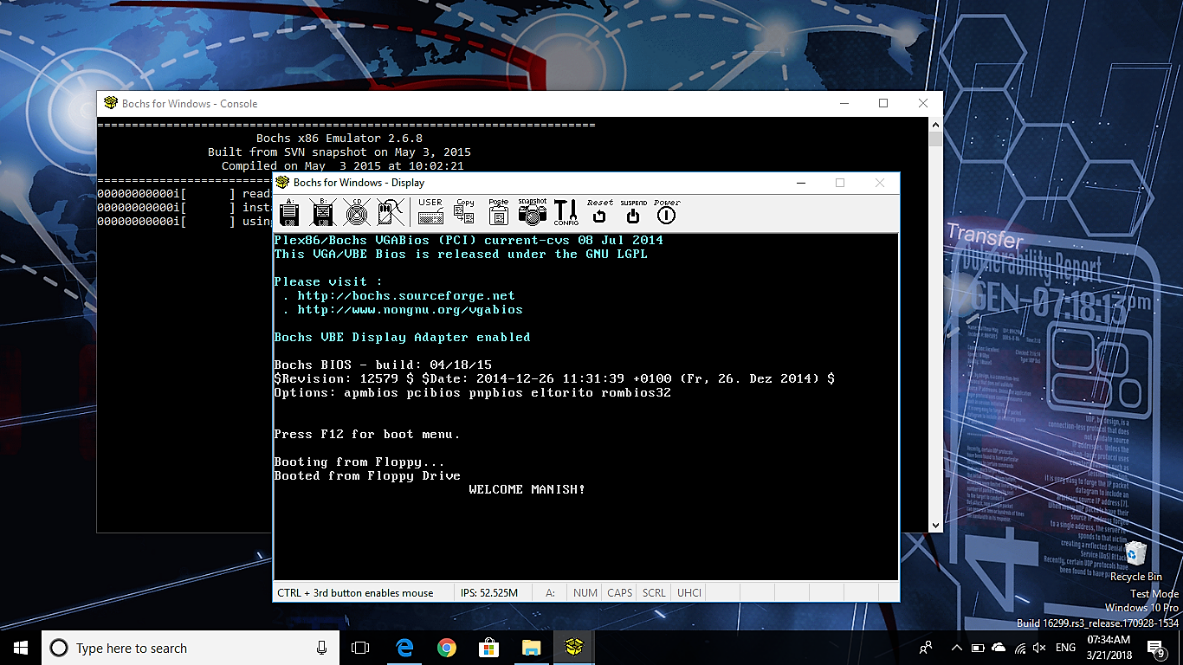
Then, prepare a batch script for building/debugging



Go to vfdwin and install a virtual drive by choosing an empty floppy img, choose its size and create. Open it.

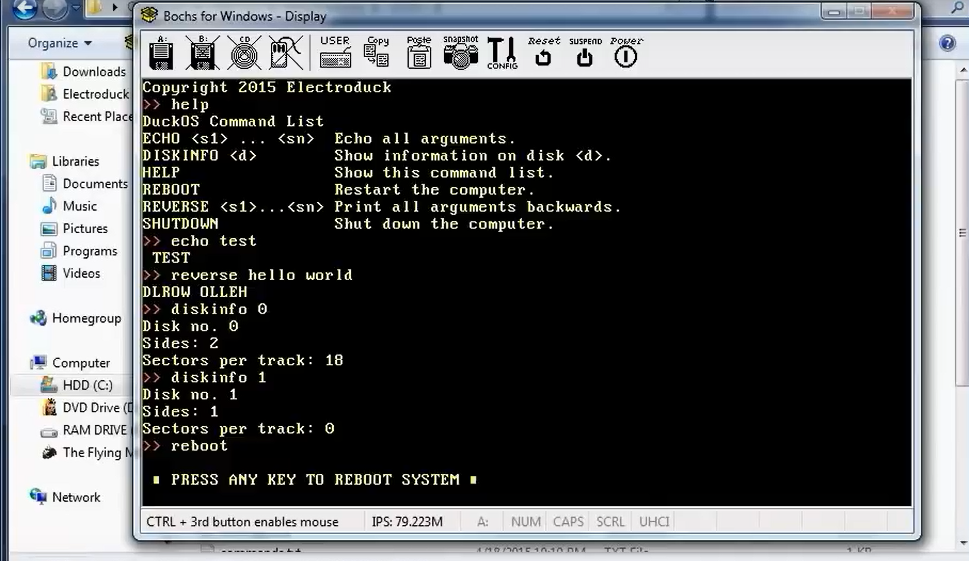
Go to build.bat and build our kernel code and it’ll automatically load kernel into floppy drive

Now,open virtual drive and our OS boots and displays message.



**Future Scope :**

This application can be easily implemented under various situations. We can add new features as and when we require. Reusability is possible as and when require in this application.There is flexibility in all the modules.

* Extensibility: It can be further extended to show disks,files,shutdown,reboot,sleep a system,authenticate with userID & password,etc.
* Reusability: Reusable software reduces design, coding and testing cost by amortizing effort over several designs. Reducing the amount of code also simplifies understanding, which increases the likelihood that the code is correct. We follow up both types of reusability: Sharing of newly written code within project and reuse of previously written code on new projects.
* Understandability: A method is understandable if someone other than creator of the method can understand the code (as well as the creator after time lapse). We use the method, which is small and coherent helps to accomplish this.
* Cost-effective: Its cost is under the budget and make within given time. It is desirable to aim for a system with a minimum cost subject to condition that it must satisfy the entire requirement. Scope of this document is to clearly identify info needed by user, the source of the information and outputs expected from the system.

**CONCLUSION:**

From a proper analysis of positive points and constraints on component, it can be safely concluded that a type of mini-OS can be created efficiently. This application is working properly and meeting to all user requirements. This component can be easily plugged in many other systems.