

# ***i*BOS**

internet Browsing Operating System

**SUBMITTED FOR THE PARTIAL FULFILMENT OF THE AWARD  
OF  
DEGREE OF B.TECH (COMPUTER SCIENCE & ENGINEERING)**

**BY**

**AMIT KHANNA ( 0009510006 )  
ANUJ GUPTA ( 0009510009 )  
MAYUR HEMANI ( 0009513022 )**

**UNDER THE GUIDENCE OF  
MR. SANJEEV PIPPAL  
H.O.D. (CS & IT)**



**DEPARTMENT OF CS & IT  
Mahatma Gandhi Mission's  
College Of Engg & Tech  
A-10, sector-62, NOIDA**

## **CERTIFICATE**

This is to certify that the project titled **iBOS** has been completed successfully in MGM's College of Engg & Tech by:

Amit Khanna ( BE-CP, Roll No. 0009510006 )  
Anuj Gupta ( BE-CP, Roll No. 0009510009 )  
Mayur Hemani ( BE-IT, Roll No. 0009513022 )

GUIDE	HOD	PRINCIPAL
(Mr. Sanjeev Pippal)	(Mr.Sanjeev Pippal)	(Mrs.Geeta Latkar)

## **ACKNOWLEDGEMENTS**

**“Outstanding achievements are not possible in vacuum. It needs lots of help and assistance besides a healthy environment, luckily we had one.”**

We take this opportunity to express our profound sense of gratitude and respect to all those who helped us throughout the duration of this project.

First and foremost we would like to thank our project guide and mentor Mr. Sanjeev Pippal, HOD, CS &IT department, for invaluable guidance throughout the duration of this project.

We would also like to thank our faculty members, Mrs Archana Saar, Mr Abhishek Garg, Mr Nitin Bajpai for their support and encouragement.

We would also thank our laboratory incharge Mr Neeraj Kaushik for his constant support and encouragement.

Last but not the least we sincerely thank our friends Akshat Gupta, Debasish Das, Sunder Singh, Shreyansh Jain, Shalini Singhi, who stood by us in testing times.

## ABSTRACT

This project aims at developing a viable web-browsing system for low-end systems. The solution is to be applied to information kiosks and networked public computers. The system is capable of providing text-based web browsing to low-end computer systems connected in a network. The system behaviorally models the popular text-based web-browser *lynx*. It is capable of supporting all functionality that a text-based hypertext system is expected to support. The system consists of two modules – the *iBOS* client, that resides at each client machine, and the *iBOS* proxy server that resides at a server machine of the network. The system also implements filtering and caching at the proxy server as well as the client. The system is designed to be low on requirements in terms of memory and disk storage. The system uses the Linux kernel (2.4/2.6), while the web-browser works as a shell to the operating system. The software is completely open in that the source codes (C++ and Java) are the actual deliverables of the project. Like any other project, this project too exhibits a number of constraints including the inability to process web-page scripts and support cookies or sessions. Finally, the system is supposed to be used as an operating system solution for public computers and kiosks.

## TABLE OF CONTENTS

Chapter No	Page No.
1. Prelude	... 9
2. Present Scenario	... 13
3. Background	... 18
4. Feasibility Study-I	... 57
5. System Requirement Specification	... 63
6. Design	...108
7. Feasibility Study– II	...118
8. Detailed Design	...124
9. Deliverables	...279
10. Conclusion and Extension	...286
11. Appendix A – References	...287
12. Appendix B – Glossary	...292
13. Index	...301

## List of Figures

<u>Figure 1 Alice Architecture</u> .....	17
<u>Figure 2 Where an OS fits in a computer system...</u> .....	Error! Bookmark not defined.
<u>Figure 3 Structuring in monolithic kernels</u> .....	Error! Bookmark not defined.
<u>Figure 4 T.H.E. system layers</u> .....	Error! Bookmark not defined.
<u>Figure 5 Client/Server model of an operating system</u> .....	Error! Bookmark not defined.
<u>Figure 6 Paging with hardware support</u> .....	Error! Bookmark not defined.
<u>Figure 7 Segmentation hardware support</u> .....	Error! Bookmark not defined.
<u>Figure 8 Basic Client Server Architecture</u> .....	Error! Bookmark not defined.
<u>Figure 9 The OSI model and the role of the browser</u> .....	Error! Bookmark not defined.
<u>Figure 10 The OSI model and the role of the browser</u> .....	Error! Bookmark not defined.
<u>Figure 11 Primary Level Use Case Diagram</u> .....	Error! Bookmark not defined.
<u>Figure 12 Secondary Level Use Case Diagram 1</u> .....	Error! Bookmark not defined.
<u>Figure 13 Secondary Level Use Case Diagram 2</u> .....	Error! Bookmark not defined.
<u>Figure 14 Tertiary Level Use Case Diagram 1</u> .....	Error! Bookmark not defined.
<u>Figure 15 Tertiary Level Use Case Diagram 2</u> .....	Error! Bookmark not defined.
<u>Figure 16 Tertiary Level Use Case Diagram 3</u> .....	Error! Bookmark not defined.
<u>Figure 17 Tertiary Level Use Case Diagram 4</u> .....	Error! Bookmark not defined.
<u>Figure 18 Integrated Use Case diagram</u> .....	Error! Bookmark not defined.
<u>Figure 19 Notation Used</u> .....	Error! Bookmark not defined.
<u>Figure 20 Dial-up model</u> .....	Error! Bookmark not defined.
<u>Figure 21 LAN-Proxy server model</u> .....	Error! Bookmark not defined.
<u>Figure 22 Thin-Client Computers</u> .....	Error! Bookmark not defined.
<u>Figure 23 iBOS Architecture</u> .....	Error! Bookmark not defined.
<u>Figure 24 BASIC IMPLEMENTATION FRAMEWORK</u> .....	Error! Bookmark not defined.
<u>Figure 25 Hierarchy of classes</u> .....	Error! Bookmark not defined.
<u>Figure 26 Network Class</u> .....	Error! Bookmark not defined.
<u>Figure 27 Server I/O Class</u> .....	Error! Bookmark not defined.
<u>Figure 28 Hierarchy of widgets</u> .....	Error! Bookmark not defined.
<u>Figure 29 Component Class</u> .....	Error! Bookmark not defined.
<u>Figure 30 Checkbox Class</u> .....	Error! Bookmark not defined.
<u>Figure 31 Image Class</u> .....	Error! Bookmark not defined.
<u>Figure 32 Radio Class</u> .....	Error! Bookmark not defined.
<u>Figure 33 Button Class</u> .....	Error! Bookmark not defined.
<u>Figure 34 Submit Button Class</u> .....	Error! Bookmark not defined.
<u>Figure 35 Reset Button Class</u> .....	Error! Bookmark not defined.
<u>Figure 36 Form Class</u> .....	Error! Bookmark not defined.
<u>Figure 37 Combo Class</u> .....	Error! Bookmark not defined.
<u>Figure 38 Hyperlink Class</u> .....	Error! Bookmark not defined.
<u>Figure 39 Textbox Class</u> .....	Error! Bookmark not defined.
<u>Figure 40 Password Class</u> .....	Error! Bookmark not defined.
<u>Figure 41 Tokenizer Class</u> .....	Error! Bookmark not defined.
<u>Figure 42 StringTokenizer Class</u> .....	Error! Bookmark not defined.
<u>Figure 43 Pass3 Tokenizer Class</u> .....	Error! Bookmark not defined.
<u>Figure 44 URL Class</u> .....	Error! Bookmark not defined.
<u>Figure 45 ActiveSerialTable Class</u> .....	Error! Bookmark not defined.
<u>Figure 46 componentList Class</u> .....	Error! Bookmark not defined.
<u>Figure 47 table Class</u> .....	Error! Bookmark not defined.
<u>Figure 48 listStack Class</u> .....	Error! Bookmark not defined.
<u>Figure 49 table engine Class</u> .....	Error! Bookmark not defined.
<u>Figure 50 Rendering Engine Class</u> .....	Error! Bookmark not defined.

<u>Figure 51 eventHandler Class</u> .....	Error! Bookmark not defined.
<u>Figure 52 Cache Class</u> .....	Error! Bookmark not defined.
<u>Figure 53 iBOS Client Handler</u> .....	Error! Bookmark not defined.
<u>Figure 54 iBOS Proxy Server</u> .....	Error! Bookmark not defined.
<u>Figure 55 Server Side Caching</u> .....	Error! Bookmark not defined.
<u>Figure 56 URL Request Class</u> .....	Error! Bookmark not defined.
<u>Figure 57. Pointers in Object diagrams</u> .....	Error! Bookmark not defined.
<u>Figure 58. Namespaces</u> .....	Error! Bookmark not defined.
<u>Figure 59. Unrelated object collection</u> .....	Error! Bookmark not defined.
<u>Figure 60. Startup time objects</u> .....	Error! Bookmark not defined.
<u>Figure 61 Global namespace objects</u> .....	Error! Bookmark not defined.
<u>Figure 62 Active Serial Table</u> .....	Error! Bookmark not defined.
<u>Figure 63 Form objects</u> .....	Error! Bookmark not defined.
<u>Figure 64 Checkbox Objects</u> .....	Error! Bookmark not defined.
<u>Figure 65 Combo Objects</u> .....	Error! Bookmark not defined.
<u>Figure 66 Hyperlink objects</u> .....	Error! Bookmark not defined.
<u>Figure 67 Image objects</u> .....	Error! Bookmark not defined.
<u>Figure 68 Password objects</u> .....	Error! Bookmark not defined.
<u>Figure 69 Radio objects</u> .....	Error! Bookmark not defined.
<u>Figure 70 Textbox objects</u> .....	Error! Bookmark not defined.
<u>Figure 71 Table objects</u> .....	Error! Bookmark not defined.
<u>Figure 72 Rendering Engine objects</u> .....	Error! Bookmark not defined.
<u>Figure 73 Proxy - Server side Objects</u> .....	Error! Bookmark not defined.
<u>Figure 74 URL Request</u> .....	Error! Bookmark not defined.
<u>Figure 75 iBOS System</u> .....	Error! Bookmark not defined.
<u>Figure 76 Browsing Subsystem</u> .....	Error! Bookmark not defined.
<u>Figure 77 AST</u> .....	Error! Bookmark not defined.
<u>Figure 78 Event Handler</u> .....	Error! Bookmark not defined.
<u>Figure 79 Pass One</u> .....	Error! Bookmark not defined.
<u>Figure 80 Table Engine</u> .....	Error! Bookmark not defined.
<u>Figure 81 Pass3 Tokenizer</u> .....	Error! Bookmark not defined.
<u>Figure 82 Global Name Space</u> .....	Error! Bookmark not defined.
<u>Figure 83 Network Subsystem</u> .....	Error! Bookmark not defined.
<u>Figure 84 Rendering Engine</u> .....	Error! Bookmark not defined.
<u>Figure 85 Libncurses</u> .....	Error! Bookmark not defined.
<u>Figure 86 System Startup</u> .....	Error! Bookmark not defined.
<u>Figure 87 General URL Request: Scenario 1 – Cache Hit</u> .....	Error! Bookmark not defined.
<u>Figure 88 General URL Request: Scenario 2 – Cache miss, proxy activated</u> .....	Error! Bookmark not defined.
<u>Figure 89 General URL Request: Scenario 3 – Proxy Failure</u> .....	Error! Bookmark not defined.
<u>Figure 90 Page Rendering Sequence</u> .....	Error! Bookmark not defined.
<u>Figure 91 Global iBOS Events: Reload</u> .....	Error! Bookmark not defined.
<u>Figure 92 Global iBOS Events: Forward</u> .....	Error! Bookmark not defined.
<u>Figure 93 Global iBOS Events: Back</u> .....	Error! Bookmark not defined.
<u>Figure 94 Widget Events: A general widget event</u> .....	Error! Bookmark not defined.
<u>Figure 95 Widget Events: Form Submission</u> .....	Error! Bookmark not defined.
<u>Figure 96 Widget Events: Hyperlink Event</u> .....	Error! Bookmark not defined.

# CHAPTER 1

## Contents

1. Prelude	
1.1 Introduction	...9
1.2 Problem Statement	...10
1.3 Scope of the project	...11



## **1.1 INTRODUCTION**

The operating system is an integral part of any computer system. Broadly speaking, it provides a user-interface, manages the system's resources, and provides an environment in which other programs can run. Owing to the growing use of the Internet, most operating systems now include a host of services that help users surf the web. However, the operating systems are meant to be general in nature. They are designed so as to provide a whole lot of services for different purposes. This renders them unsuitable for computers dedicated to specific applications such as browsing the Internet. They are simply too cluttered up with programs that are often of no use to an Internet-user. This makes the operating system too heavy. The following project is an endeavor to develop an Internet browser that requires no separate operating system. The idea is to integrate all necessary operating system features in the browser itself, so that it runs on a standalone computer without any additional overheads. The system comprises of a graphical Internet user-interface and an integrated kernel and file-system. The primary features of this Internet-browsing software system are a friendly Windows-Menu-Pointing device interface, a web-browser, and programs that control the different resource-management aspects. The final project is aimed at a viable solution for E-Commerce kiosks, and Internet-browsing computers.

## 1.2 PROBLEM STATEMENT

The world-wide web is an interconnected collection of hypermedia documents. To access www-documents users need a client software called web-browser. Some of the popular web browsers used are Internet Explorer, Netscape Navigator etc. Most contemporary operating systems now support web-browser software. However, when the sole use of a computer is to surf and use the web, conventional operating systems tend to be unsuitable. With respect to conventional Internet-capable operating systems and in an environment where web-surfing is the single most important function, the following are the major problems: -

- Most conventional operating systems run a host of services that are never required for web-browsing. This may sometimes prove to be a bottleneck for productivity.
- Most of the operating systems demand heavy hardware resources, for example, memory.
- Most of the operating systems are cluttered with services which makes them too heavy. This makes them unsuitable for lower end machines such as 486 which can otherwise be used easily for a specific application such as surfing the web.
- Most web-browsers allow downloading of web-resources. This poses a certain degree of security threat to the host system.
- Most operating system allows direct access to the files stored on a computer. This renders such systems unsuitable for use as public computers, or kiosks.

The present project is an attempt to provide a viable solution which addresses the above stated issues. It aims at designing and developing an operating system called iBOS (Internet Browsing Operating System) capable of browsing the web.

## 1.3 SCOPE OF THE PROJECT

Operating systems are the resource managers of computer systems. Contemporary operating systems provide ample support for using the Internet services. The most commonly used Internet service is the world-wide-web browsing service. Most operating systems are not dedicated to browsing the web, resulting in the base software to be heavy for low-end machines. Also, this often results in poor user-performance.

The present project aims at developing a web-browser shell that is supported on a light-weight operating system. The scope of the project can be summarized in the following points: -

- The final product aims at a viable solution for standalone or networked e-commerce workstations.
- The programs that the system is presently aimed at providing includes:
  - ♣ A text-based web-browser capable of laying out HTML pages
  - ♣ An efficient HTTP client
  - ♣ An elementary operating system to handle the different house-keeping functions
- The user-interface is based on the notion of Windows-Icons-Menus-Pointing devices interfaces.
- The bare minimum requirements for the hardware which could run the system are:
  - ♣ Intel x86 processor (386+)
  - ♣ At least 16 MB RAM
  - ♣ A few hundred megabytes of hard-disk space
  - ♣ VGA/SVGA graphics-capabilities
  - ♣ Console I/O devices – VDU, Keyboard, and a mouse
- With respect to functionality and support, the following things are NOT supported as for now:
  - ♣ JavaScript (or other scripts), Java, Web-animations (Shockwave etc.) and multimedia-content in web-pages are not supported.
  - ♣ Styling documents, XML, CSS etc. are not supported

- ♣ There are no extra device-drivers. The only devices that are supported are the bare-minimum for any GUI, plus the network device drivers. USB devices, floppy disks, CDROM-drives are also not supported.
- ♣ A user is not allowed to store web-pages and other web-resources on the hard-disk. This feature is not supported because the software is meant for public computers where such a capability can pose a threat.

However, these missing capabilities can be added into the system because the development for this system can be done using any ANSI – C compiler on a Linux system.

- A basic operating system provides the general house-keeping services, including
  - ♣ Memory management
  - ♣ Process management
  - ♣ I/O management
  - ♣ Virtual-memory management

For providing the operating system functions efficiently, the kernel of any standard operating system (viz. Linux, Minix etc.) can be used.

## **CHAPTER 2**

### **Contents**

#### 2. Present Scenario

2.1 Introduction ...14

2.2 Case Study ... 16

## **2 PRESENT SCENARIO**

### **2.1 Introduction**

In the present world not only information but also easy access to information is an important issue. The Internet provides probably the biggest and the most comprehensive source of information related with any field. The growing need and importance of the Internet in today's world gave birth to 'Internet kiosks'. Internet kiosks are simple machines installed at various public places such as libraries, galleries and museums, educational institutions, municipal offices, visitor information booths, conferences, shopping malls, airports--in short any location where simple, easy access to information is desirable.

These machines provide the end user with the facility of surfing the web with ease. Since these machines are designed to provide limited functionality and are dedicated for specific applications, the manner in which the underlying software or to be specific the operating system is designed is different from the usual approach. A generic operating system provides a host of services to enable a wide range of applications which makes it heavier. But while designing operating system for kiosks, a special care is taken to provide only those bare minimum functionalities that are needed to support the end application.

There are many kinds of approaches to developing Internet kiosks, from expensive solutions based on proprietary software to HTML-based open-source solutions. Browser-based technologies are particularly attractive as they are inherently multimedia, and the text, graphics, sound and streaming media, and the content offered by them is highly portable.

Linux is one of the most preferred platform for kiosks development mainly attributed to its flexibility. Also Linux is inexpensive to set up and offers a wide range of options, from diskless display stations to self-contained, database-driven web servers. Due to the versatility of the underlying operating system, a well-designed Linux kiosk can be easily deployed at a remote location and administered via a telephone or network link.

Setting up Linux to be used as a web-based kiosk is some what similar to setting up Linux as a graphical workstation, with a few additional considerations. If the kiosk is a standalone unit, it boots straight into kiosk mode. Also, the browser environment is modified to disable standard dialog screens and eliminate parts of the browser that allow inappropriate interaction, such as saving the file option. Provisions are also made to prevent improper use of Internet kiosk by restricting the content that a user can view.

A Linux kiosk is often connected back to home site via a direct network or PPP link, requesting pages from the home site, or is a self-contained website, running Apache and any corresponding CGI modules.

An important issue is modifying HTML pages so as to make them better suited for Internet Kiosks. This is mainly due to the fact that using a kiosk is a different experience from sitting at a desk browsing the web from a personal workstation. Some of the important considerations while designing such web pages are:

- Removing all external A HREF' s, including MAILTO' s.
- Removing all unnecessary verbiage, logos, etc. from the pages.
- Increasing font sizes if necessary.
- Removing unnecessary and unwanted animations.
- Disabling pop ups.
- Keeping text passages very short and using large fonts.
- Avoiding screens that require scrolling.
- Creating embedded navigation, e.g., return to previous page, go to next page, go to kiosk menu, etc., all designed in the pages themselves.
- Disabling sound and video files.
- Removing animated GIF' s.

## 2.2 Case study:

A number of operating systems have been developed that are designed specifically for supporting Internet browsing only. One of the most popular Internet browsing enabled operating system is Alice. Alice is an embedded operating system. It is used extensively for hand-held computers, PDA etc. Alice is highly efficient in displaying HTML pages in a resource-limited environment.

Alice is not a cut-down version of a desktop browser. It has been designed from scratch for embedded systems. Its compact, modular architecture makes it easy for OEMs to customize to work on a particular hardware.

Some of the major features of Alice operating system are:

- Compact: requires less than 200 Kb ROM and 400 Kb RAM.
- Modular: display driver and input device drivers are separately loaded modules, making it easy to port to non-standard hardware.
- No operating system required when using the optional file system and bootstrap loader modules.
- Precompiled versions for DOS available.
- Windows 95 emulator makes it easy to see how HTML will look on the target device.
- Display resolutions down to 1/4 VGA in monochrome, grayscale or color.
- Range of fonts including serif (similar to Times Roman) and sans-serif (similar to Arial). Others are available to order.
- Supports the most commonly used HTML 4.01 tags, including frames, tables and forms.
- Displays GIF and JPEG images.
- HTTP 1.0
- Integrated TCP/IP stack makes it easy to configure and deploy.
- Dialer and PPP supplied for dial-up connection.

### System Requirements

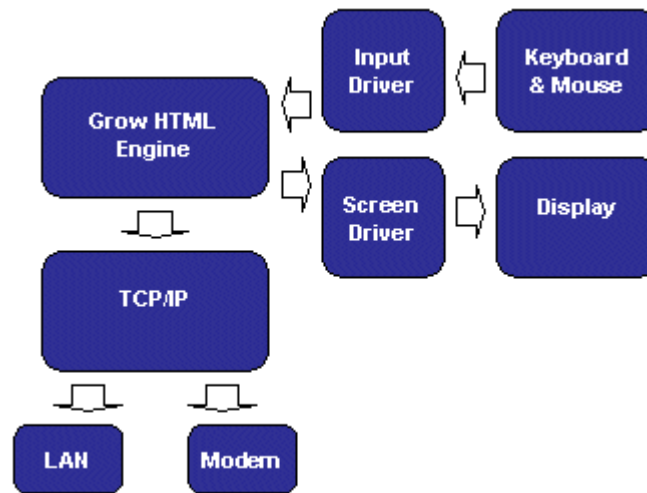


For DOS version

- Intel x86 compatible CPU
- 400 Kbytes free RAM
- MS-DOS 3.3 or compatible
- A network adapter and packet driver or Hayes compatible modem.

For no operating system version it provides a host of options including support for flash and disk file systems, Ethernet drivers and so on.

### Basic Architecture



**Figure 1 Alice Architecture**

Alice is based on the Grow Graphical User Interface which means GUI is loaded as and when required. It uses object oriented techniques to make it easy to change the "personality" to suit the target environment, for example to look like Windows or Mac. It dynamically loads driver files needed for the device-dependent parts of the system such as the display and input devices. It comes with a standard set of drivers.

### Screen Drivers

- VGAHI.GDL - VGA 640 x 480 in 256 colors
- VGABW.GDL - VGA 640 x 480, black and white
- CGA.GDL - CGA 320 x 200 in 16 colors

### Input Drivers

- MSMOUSE.GDL - MS compatible mouse
- NOMOUSE.GDL - Dummy driver

## **CHAPTER 3.**

### **Contents**

#### **3. Background**

##### **3.1 Operating System**

3.1.1 Introduction	...19
3.1.2 Basic Operating System Functions	...21
3.1.3 Operating System Architectures	...23
3.1.4 Operating System Design Issues	...26

##### **3.2 Web Browsers**

3.2.1 World Wide Web	...35
3.2.2 Hypertext Protocol-HTTP/1.0	...35
3.2.3 HTML	...36
3.2.4 Uniform Resource Locator	...36
3.2.5 Web browser Software	...37
3.2.6 Web Browser Concepts	
3.2.6.1 Protocol Stack	...38
3.2.6.2 Domain Name Server	...39
3.2.6.3 Caching	...40
3.2.7 Components of a Web Browser	
3.2.7.1 HTTP Client	...44
3.2.7.2 Layout Engine	...50
3.2.7.3 Rendering Engine	...52
3.2.8 Working of a Web Browser	...55