

Major Project Report on

Software Installer using Client Server Module

Submitted By:

Rishabh Chaudhary (14MI508)

Akhilesh Kumar (14MI526)

Rahul Kumar (14MI538)

Gourav Sisodiya (14MI560)

Under supervision of:

Dr. T.P. Sharma

Associate Professor

NIT Hamirpur



Computer Science and Engineering Department

National Institute Of Technology, Hamirpur

December 2017

Declaration

We are, hereby, declare that this written submission represents our ideas in my own words and where other's ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be the cause of disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Rishabh Chaudhary
(14MI508)

Akhilesh Kumar
(14MI526)

Rahul Kumar
(14MI538)

Gourav Sisodiya
(14MI560)

Date:

13-12-2017

ACKNOWLEDGEMENT

We take immense pleasure in thanking National Institute of Technology, Hamirpur for providing us the great chance for learning and development. Therefore, we consider ourselves very lucky being a part of this project.

We consider ourselves privileged to express gratitude and respect towards all those who guided us through this project. It is with hearty gratitude that we acknowledge their contributions to our project.

We would like to express our sincere gratitude and heartfelt thanks to “Dr. T. P. Sharma, Assistant Professor Computer Science and Engineering Department of NIT Hamirpur” for his unflinching support and guidance, valuable suggestions and expert advice. His words of wisdom and expertise in subject matter were of immense help throughout the duration of this project. He has been a true mentor and without his support and guidance, the completion of this project would not have been possible.

We perceive this opportunity as a big milestone in my career development. We will strive to use gained skills and knowledge in the best possible way, and We will continue to work on their improvement. Hope to continue cooperation with all of you in the future.

ABSTRACT

System administrator has to install and update software on every computers regularly. This job is very tiring and time consuming. The aim of this project is to make a software which will help the admin to update or install softwares of every computers connected to his single network by issuing a command through his computer

This software allows the admin to create a file of commands which will be broadcast to all other computers. Then those computers will extract the commands from the file received and then execute them one by one.

This project is developed by using Python and socket programming for communicating with other computers in the network.

LIST OF FIGURES

| Figures | Page No. |
|----------------------------|-----------------|
| 7.1 Use Case Diagram..... | 16 |
| 7.2 Flowchart..... | 17 |
| 7.3 Data Flow Diagram..... | 18 |

CONTENTS

1. Introduction

| | | |
|-----|----------------|---|
| 1.1 | Overview..... | 8 |
| 1.2 | Scope..... | 8 |
| 1.3 | Objective..... | 8 |

2. Technologies used

| | | |
|-----|-------------------------|---|
| 2.1 | Python..... | 9 |
| 2.2 | Socket Programming..... | 9 |

3. Requirements

| | | |
|-----|----------------------------------|----|
| 3.1 | Functional Requirements..... | 10 |
| 3.2 | Non-Functional Requirements..... | 10 |

4. System Analysis.....11

| | | |
|-----|----------------------|----|
| 4.1 | Existing System..... | 11 |
| 4.2 | Proposed System..... | 11 |

5. Feasibility Study.....12

| | | |
|-----|------------------------------|----|
| 5.1 | Technical Feasibility | 12 |
| 5.2 | Economical Feasibility | 12 |
| 5.3 | Operational Feasibility..... | 12 |

6. Terminologies Used13

7. System Design and Diagrams.....15

| | | |
|-----|------------------------|----|
| 7.1 | Admin Section..... | 15 |
| 7.2 | Host Section..... | 15 |
| 7.3 | Use Case Diagram..... | 15 |
| 7.4 | Flowchart..... | 16 |
| 7.5 | Data Flow Diagram..... | 17 |

| | |
|----------------------------------|----|
| 8.Implementation | 19 |
| 8.1 Introduction | 19 |
| 8.2User Training | 19 |
| 8.3Security and Maintenance..... | 19 |
| 9.Conclusion | 21 |

Chapter 1. Introduction

1.1 Overview

This project named software installer, is an open source software. It will allow the system administrator of the lab to updating and installing the software in the lab from a single computer.

1.2 Scope

This software can be used in any computers connected to a single network like a lab. The only constraint is that they all must run ubuntu with python installed.

1.3 Objective

Usually the admin has to install or update the softwares in each and every computers in the lab. This task is very tiring and cumbersome. We have developed this software to get rid of this. By using our program admin can issue a single or multiple commands which will be executed in the rest of the systems in the lab. The admin has to write those commands in a single file and then our software will execute them.

Chapter 2. Technologies Used

2.1 Python: Python is a high level programming language that lets you work quickly and integrate systems more effectively. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** – Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- IT supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

2.2 Socket programming: Socket programming is a way of connecting two nodes or systems on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while other socket reaches out to the other to form a connection. Server forms the listener socket while client reaches out to the server.

The processes that use a socket can reside on the same system or different systems on different networks. Sockets are useful for both stand-alone and network applications. Sockets allow you to exchange information between processes on the same machine or across a network, distribute work to the most efficient machine, and they easily allow access to centralized data. Socket application program interfaces (APIs) are the network standard for TCP/IP.

Chapter 3. Requirements

3.1 Introduction

A software requirements specification (SRS) is a comprehensive description of the intended purpose and environment for software under development. The SRS fully describes what the software will do and how it will be expected to perform.

An SRS minimizes the time and effort required by developers to achieve desired goals and also minimizes the development cost. A good SRS defines how an application will interact with system hardware, other programs and human users in a wide variety of real-world situations. Parameters such as operating speed, response time, availability, portability, maintainability, security and speed of recovery from adverse events are evaluated.

3.2 Functional Requirements

- Less time consuming
- Easy to implement and handle
- Backup data can easily be generated
- Data redundancy can be avoided to a great extent
- Very less manual work

3.3 Non Functional Requirements

1. Security: The system back-end server shall only be accessible to authenticated administrators.
2. Availability: The system should be available whenever code needs to be executed and only restricted by the down time of the server on which the system runs.
3. Maintainability: The authenticated server takes care of the code execution. In case of a failure, a re-initialization of the program will be done.

Chapter 4. System Analysis

4.1 Existing System

In the existing system, the installations of any software on the systems in the lab is done manually which is very time consuming for the students as well as for the lab assistants. The installation process on every system is very chaotic and students or lab assistants have to manually install the software which is not a very feasible solution .

In existing system, the process of the manual installation of softwares is time consuming and chaotic thus a new system is proposed.

4.2 Proposed System

The proposed system is having many advantages over the existing system. It requires less overhead and is very efficient. The proposed system deals with the installation process efficiently because everything is done online from a dedicated server and server can run commands on remote hosts and execute the code on the hosts itself, thus making the new proposed system more convenient and user friendly.

Some of the advantages of new system over existing system:

- Less time consuming
- More user friendly
- More convenient
- More transparent as compared to existing system
- Hassle free and chaos free

Chapter 5. Feasibility Study

5.1 Technical feasibility

Technical feasibility assesses the current resources (such as hardware and software) and technology, which are required to accomplish user requirements in the software within the allocated time and budget. The technical feasibility in the proposed system deals with the technology used in the system. It deals with the hardware and software used in the system whether they are of latest technology or not. It happens that after a system is prepared a new technology arises and the user wants the system based on that technology. This system uses Ubuntu and Python 3.x.

5.2 Economical feasibility

This system doesn't require any extra hardware and it is run on college server, it is necessary to consider the benefits that can be achieved by developing the software so we can say that it is also economic feasible.

5.3 Operational feasibility

Operational feasibility assesses the extent to which the required software performs a series of steps to solve business problems and user requirements. The project has been developed in such a way that it becomes very easy even for a person to operate it. This software is very user friendly and does not require much technical person to operate. The user should only have knowledge of basics of linux and its commands and python (basics). Thus the project is even operationally feasible.

Chapter 6. System Design

Server

A server is a computer program that processes requests, provides services and deliver data to other computer programs either on same or other computers. The system in which server program runs is also referred to as a server. That machine may be only a dedicated server or can be used for other purposes too.

Client-server systems are today most frequently implemented by (and often identified with) the request-response model: a client sends a request to the server, which performs some action and sends a response back to the client, typically with a result or acknowledgement. Designating a computer as "server-class hardware" implies that it is specialized for running servers on it

Client

A client is a computer program that, depends on sending a request to another computer program (that may or may not be located on another computer).

In other words a client is a piece of computer hardware or software that accesses a service made available by a server. The server is often (but not always) on another computer system, in which case the client accesses the service by way of a network. The term applies to the role that programs or devices play in the client-server model.

Multicast

Multicast is the term used to describe a group communication when some information is sent from one or more points to a finite set of other points. In multicast ,there may be one or more senders, and the information is distributed to a set of receivers.

Group communication may either be application layer multicast or network assisted multicast, where the latter makes it possible for the source to efficiently send to the group in a single transmission. Copies are automatically created in other network elements, such as routers, switches and cellular network base stations, but only to network segments that currently contain members of the group.

Unicast

The term Unicast is used to describe communication where information is sent from a point to another point. In case of unicast, there is just one receiver, and one sender each identified by a network addresses.

Unicast transmission, in which a packet is sent from a single source to a specified destination, is still the predominant form of transmission on LANs and within the Internet. All LANs (e.g.

Ethernet) and IP networks support the unicast transfer mode, and most users are familiar with the standard unicast applications (e.g. http, smtp, ftp and telnet) which employ the TCP transport protocol.

Broadcast

In general, to broadcast (verb) is to cast or throw forth something in all directions at the same time. A radio or television broadcast (noun) is a program that is transmitted over airwaves for public reception by anyone with a receiver tuned to the right signal channel.

The term Broadcast is used to describe communication where information is sent from one point to all points. In broadcast there is just one sender, but the information is sent to all connected receivers.

Socket

A network socket is an internal endpoint for sending or receiving data at a single node in a computer network. We can also see it as a combination of IP address and port.

Port

A port is an endpoint of communication in an operating system. A socket is a software object that acts as an end point establishing a bidirectional network communication link between a server-side and a client-side program.

In UNIX, a socket can also be referred to as an endpoint for interprocess communication(IPC) within the operating system(OS).

UDP

User Datagram Protocol(UDP) is an unreliable protocol used for communication. There is no guarantee that packets will be delivered. But it is faster than its reliable counterpart TCP (Transmission Control Protocol).

Chat Server

A chat server is a computer dedicated to providing the processing power to handle and maintain chatting and its users.

Chapter 7. System Design and Diagrams

The system design is mainly divided into two main sections:

7.1 Admin Section

Admin runs the server side of our program. Admin has to put all commands to install the software or any other command that he requires to execute in the host.

7.2 Host

Host part of the software has the responsibility of executing the commands send by the admin. It constantly listens to a specified port and when it receives the file sent by the admin which contains the commands to execute, it reads the file and executes the commands one by one.

Diagrams

7.3 Use Case Diagrams

Use case diagrams model behavior within a system and helps the developers understand of what the user require. The stick man represents what's called an actor. Use case diagram can be useful for getting an overall view of the system and clarifying who can do and more importantly what they can't do. Use case diagram consists of use cases and actors and shows the interaction between the use case and actors.

1. The purpose is to show the interactions between the use case and actor.
2. To represent the system requirements from user's perspective.
3. An actor could be the end-user of the system or an external system.

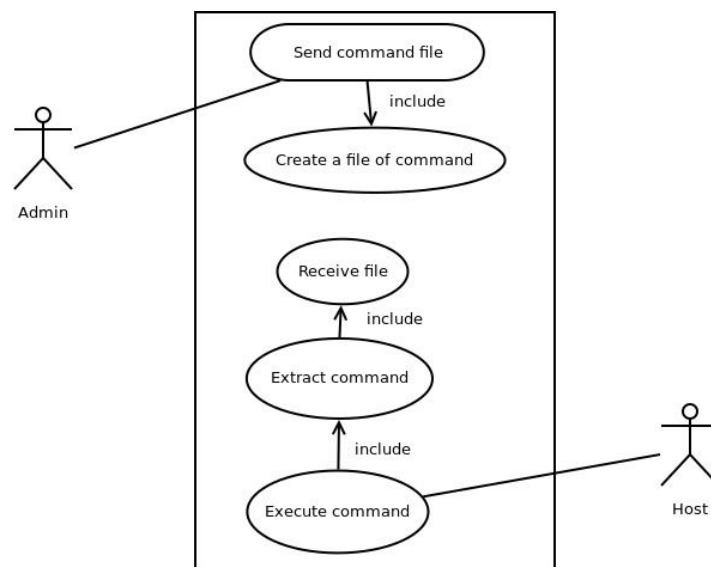


Fig. 7.1 Use Case Diagram

7.4 Flowchart

A flowchart is a formalized graphical representation of the process structure or system, logic sequence, algorithm or the structural solution of the problem. The Flowchart basically describes the flow of data through an information processing systems.

The flowchart purpose is to provide people with a general language or reference point when working with a project or process. The flow is a set of the logical sequence of operations that fulfill the certain requirements.

A Flowchart helps to see that how the work of the process can be improved, and also allows to find the key elements of the process and remove the steps that are not required or even excessive.

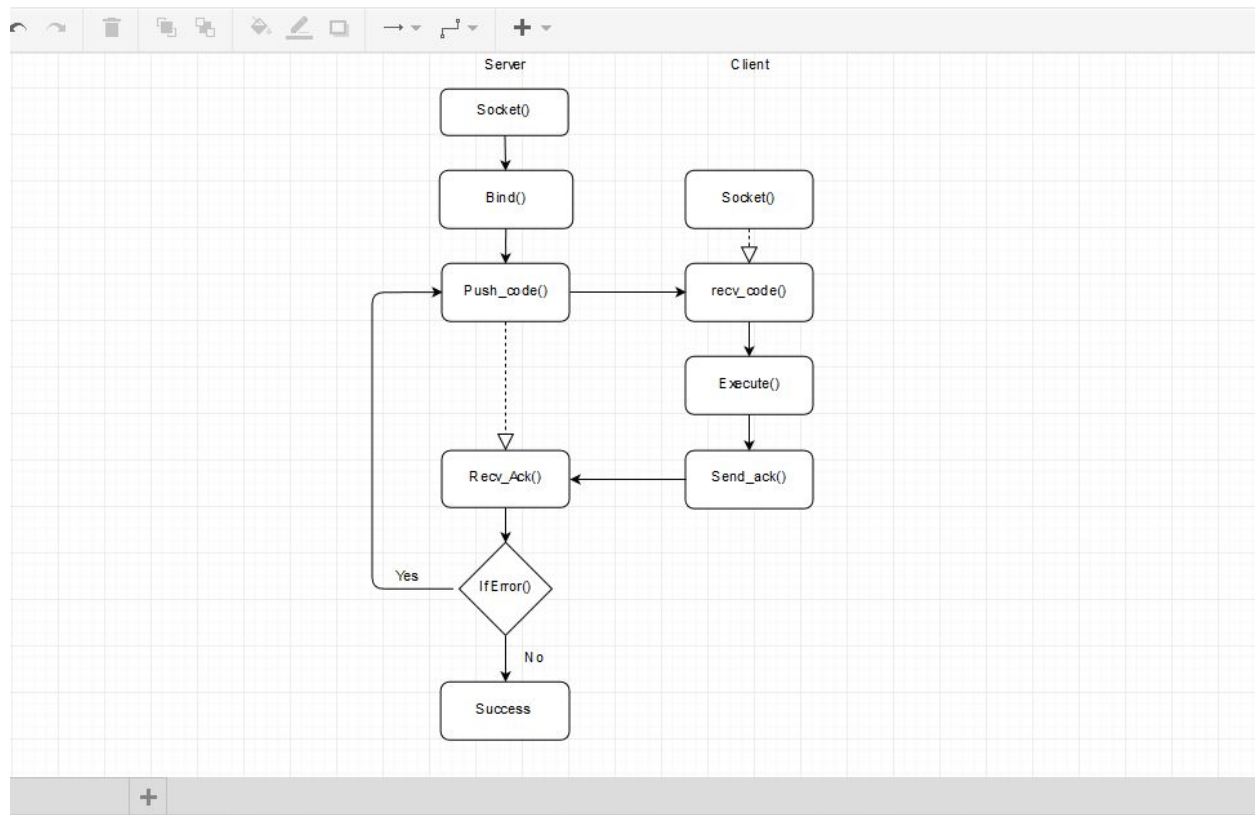


Fig. 7.2 Flow Chart

7.5 Data Flow Diagram

Data flow diagram (DFD) represents how data is processed by a system and its flow between different processes in a system. It is a graphical method that describes flow of information and the transforms or functions that are applied as data move from input to output. It provides a simple, interactive method for describing processes without much focusing on the details of computer system. DFDs are user-friendly technique because they provide details about what users do rather than what computers do.

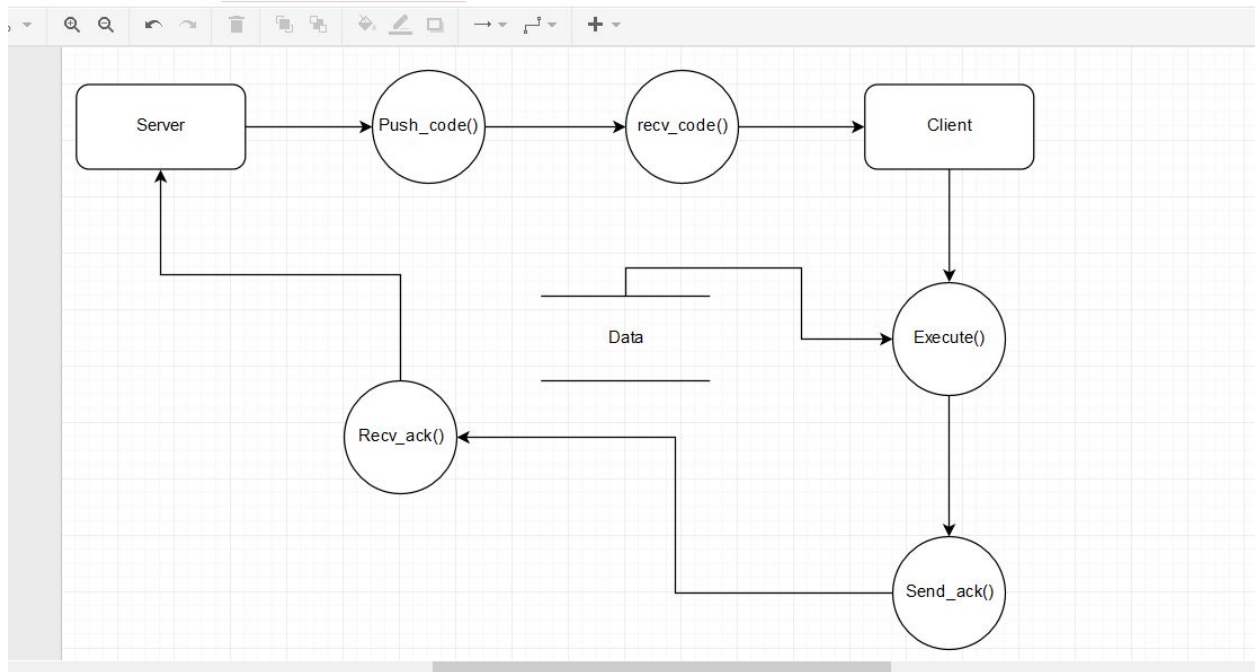


Fig. 7.3 Data Flow Diagram

Chapter 8. Implementation

8.1 Introduction

Implementation is the stage in the project where the theoretical design is turned into a working system and is giving confidence on the new system for the users that it will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the change over, an evaluation of changed over methods. Apart from planning major task of preparing the implementation are education and training of users, the implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out, discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system. In network backup system no additional resources are needed.

Implementation is the final and the most important phase. The most critical stage in achieving a successful new system is giving the users confidence that the new system will work and be effective. The system can be implemented only after thorough testing is done and if it is found to be working according to the specification. This method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain type of transactions while using the new system.

8.2 User training

After the system is implemented successfully, training of the user is one of the most important sub tasks of the developer. For this purpose user manuals are prepared and handed over to the user to operate the developed system. Thus the users are trained to operate the developed system. Both the hardware and software securities are made to run the developed systems successfully in future. In order to put new application system into use, the following activities were taken care of:

- Preparation of user and system documentation.
- Conducting user training with demo and hands on.
- Test run for some period to ensure smooth switching over the system.

The users are trained to use the newly developed functions. User manuals describing the procedures for using the functions listed on menu are circulated to all the users. It is confirmed that the system is implemented up to users need and expectations.

8.3 Security and Maintenance

Maintenance involves the software industry captive, typing up system resources .It means restoring something to its original condition. Maintenance follows conversion to the extent that

changes are necessary to maintain satisfactory operations relative to changes in the user's environment. Maintenance often includes minor enhancements or corrections to problems that surface in the system's operation. Maintenance is also done based on fixing the problems reported, changing the interface with other software or hardware enhancing the software.

Any system developed should be secured and protected against possible hazards. Security measures are provided to prevent unauthorized access of the database at various levels. An uninterrupted power supply should be so that the power failure or voltage fluctuations will not erase the data in the files. Password protection and simple procedures to prevent the unauthorized access are provided to the users .The system allows the user to enter the system only through proper username and password.

Chapter 9. CONCLUSION

We conclude that our software can be used to install or update software in the computers connected in a network. To do this the admin need to create a file containing the commands to update or install the software and then using our software she need to broadcast it. The computers then extracts the commands from the file then executes them one by one.

Our software is also very easy to use. User has to only broadcast the file containing the command through our software, then our software will take care of the rest. The user must have a basic knowledge of linux command and how to run a python program that's all.