ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

Continuous Regression Test Framework

Rakshith KN[#1], Dr. T H Sreenivas[#2]

PG Student, Department of IS&E, The National Institute of Engineering, Mysore, India[#1]

Professor, Department of IS&E, The National Institute of Engineering, Mysore, India[#2]

ABSTRACT: Today Power Systems support following Linux distributions, Red Hat Enterprise Linux (RHEL), Ubuntu, and SUSE Linux Enterprise Server (SLES). These Linux flavors are supported in various operating environments As a Power KVM Guest, As a Power VM Guest and Running in non-virtualized environment (bare-metal mode). Linux on Power test teams within Linux Technology Center are facing a problem with exponentially growing Hardware & Software combinations, Expanding Hardware platforms, and Exploding combinations of Hardware, Firmware and Software. In addition the test teams are required to frequently validate several generally available Linux distributions on newer platforms and hardware. To overcome these challenges and difficulties an automated regression test framework is being developed with a simple Goal - End to End automated continuous regression test framework to facilitate optimal testing of multiple flavors of Linux (Host and Guest OS) under different Firmware levels & Operating / Power Hardware environments.

KEYWORDS : LTS, Non – LTS, Build Availability, Test Preparation, Test Execution.

I. INTRODUCTION

Linux on Power test teams within Linux Technology Center are facing a problem with exponentially growing Hardware & Software combinations, Expanding Hardware platforms, and Exploding combinations of Hardware, Firmware and Software. In addition the test teams are required to frequently validate several generally available Linux distributions on newer platforms and hardware. To overcome these challenges and difficulties an automated regression test framework is being developed with a simple Goal - End to End automated continuous regression test framework. The Project is titled Continuous Regression Test Framework as we continuously check for the Various Releases of Linux Which Can be either LTS or NON LTS, Once we have these releases we download both the Iso images and NetBoot files which are later installed on the Hardware in following three categories namely Bare Metal ,POWER VM and POWER KVM. Bare Metal is a direct installation of OS on the Hardware and the other two POWER VM and POWER KVM are the virtual Environments of which the former is software Level Virtualization and later is a Hardware level virtualization. Once we have this Os on the system and the system is up and running we Perform Regression Testing to ensure that all test cases are without any errors and accepted by the end user of the Systems. All these activities are integrated into Jenkins server for deployment and start executing automatically when there is a release for particular Linux release and generate results which are displayed to the user and a mail will be sent to the Jenkins mailing list. The necessity was to have a efficient and effective testing framework that would reduce the problem faced in the growing software and hardware combinations with respect to the Server Testing for the specified work. As a result of this project Continuous Regression Test Framework allows Test team to test various releases of Linux Distributions on the Machines at a better quality level at low cost which is critical in pay for performance environment. This solution reduces the cost of overall manual testing which includes lot of man hours and manual interventions. Also manual testing would take lot of execution time and ineffective resource utilization this made the way for automation of the test execution for testing on the Servers under different operating environments with various combination of hardware and software. It also makes efficient usage of available resources and results in better utilizations of the resources available. This Continuous Regression Test Framework helps customers experience the latest available release of the Linux Distributions.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

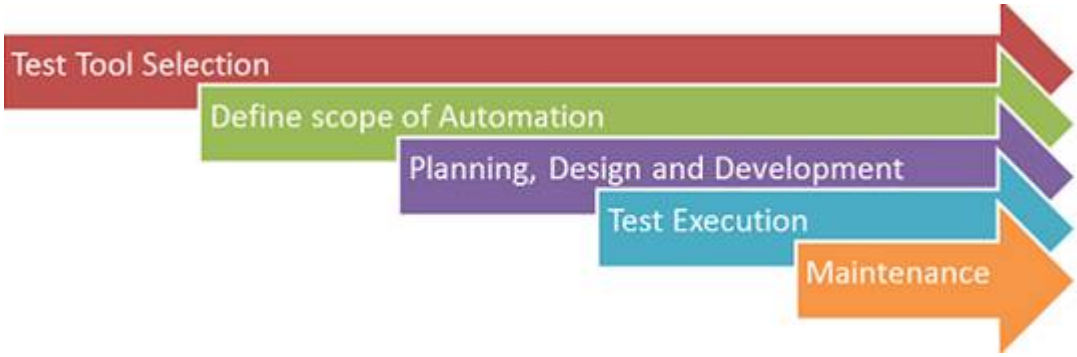


Figure 1: Automation Process

- Test Tool Selection : Test tool selection plays a very prominent role with in the organization, mainly because it will be effective in solving many problems both effectively and efficiently. It should be ready with the changes coming in the path.
- Define scope of automation : Scope of automation is the area of your Application Under Test which will be automated. Following points help determine scope, feature that are important for the business, Scenarios which have large amount of data, Common functionalitiesacross applications, Ability to use the same test cases for cross browser testing.
- Planning, design and Development : During this phase you create Automation strategy & plan, which contains following details-Automation tools selected, framework design and its features, in-Scope and Out-of-scope items of automation, automation test bed preparation, schedule and Timeline of scripting and execution, deliverables of automation testing
- Maintenance : As new functionalities are added to the System Under Test with successive cycles, Automation Scripts need to be added, reviewed and maintained for each release cycle. Maintenance becomes necessary to improve effectiveness of Automation Scripts. [1]

Selecting the right tool can be a tricky task. Following criterion will help you select the best tool for your requirement-

- Environment Support
- Ease of use
- Testing of Database
- Object identification
- Image Testing
- Error Recovery Testing
- Object Mapping
- Scripting Language Used
- Support for various types of test - including functional, test management, mobile, etc...
- Support for multiple testing frameworks
- Easy to debug the automation software scripts
- Ability to recognize objects in any environment
- Extensive test reports and results
- Minimize training cost of selected tools

II. DESIGN

Design is a phase where in the internal logic of each of the module that explains in detail with the help of flowcharts. It includes details and algorithmic design of each of the modules is specified [2]. Each subsection provides an complete

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

idea in detail about some of the important modules. It also describes about the flow of control in the software with description about software modules by explaining the details about each of the components [3].
It also explains the following:

1) Each module with the flow control.
2) Modules with component description.

Structure chart shows the control flow among the modules in the system, explains the identified modules and the interaction between the modules. It also explains the identified sub modules. Structure chart also explains the input for each modules and output generated by each module. The various key components identified are:

a) Build Availability
b) Test Preparation
c) Test Execution

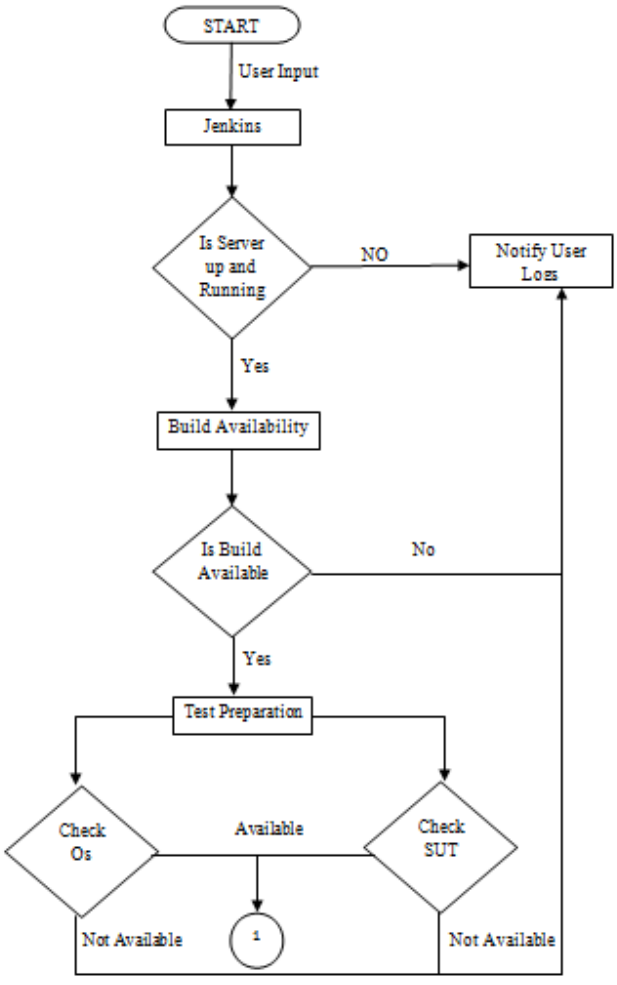


Figure 2: Flow Diagram

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

III.BUILD AVAILABILITY

Purpose: The purpose of this module is to check for the Availability of a Particular Linux Release. The Linux Release can be LTS release or the Non LTS which are Long Term Support and Non Long Term Support respectively. Once the Build is available the Particular release will be downloaded and placed in a repository whose download path will be specified in the code. It checks for the updates in the release for every week

Functionality: Enables user to potentially layer the functionality of the application and give the freedom to the user to specify what Linux Flavour to be downloaded, i.e. Rhel, Sles, and Ubuntu.

Input: The input to this module can be any flavour of Linux that need to be downloaded. They can even specify whether to download iso image or the net-boot files or both.

Output: The output of this module will be a Linux release that will be downloaded in a particular repository in a format specified by the user.

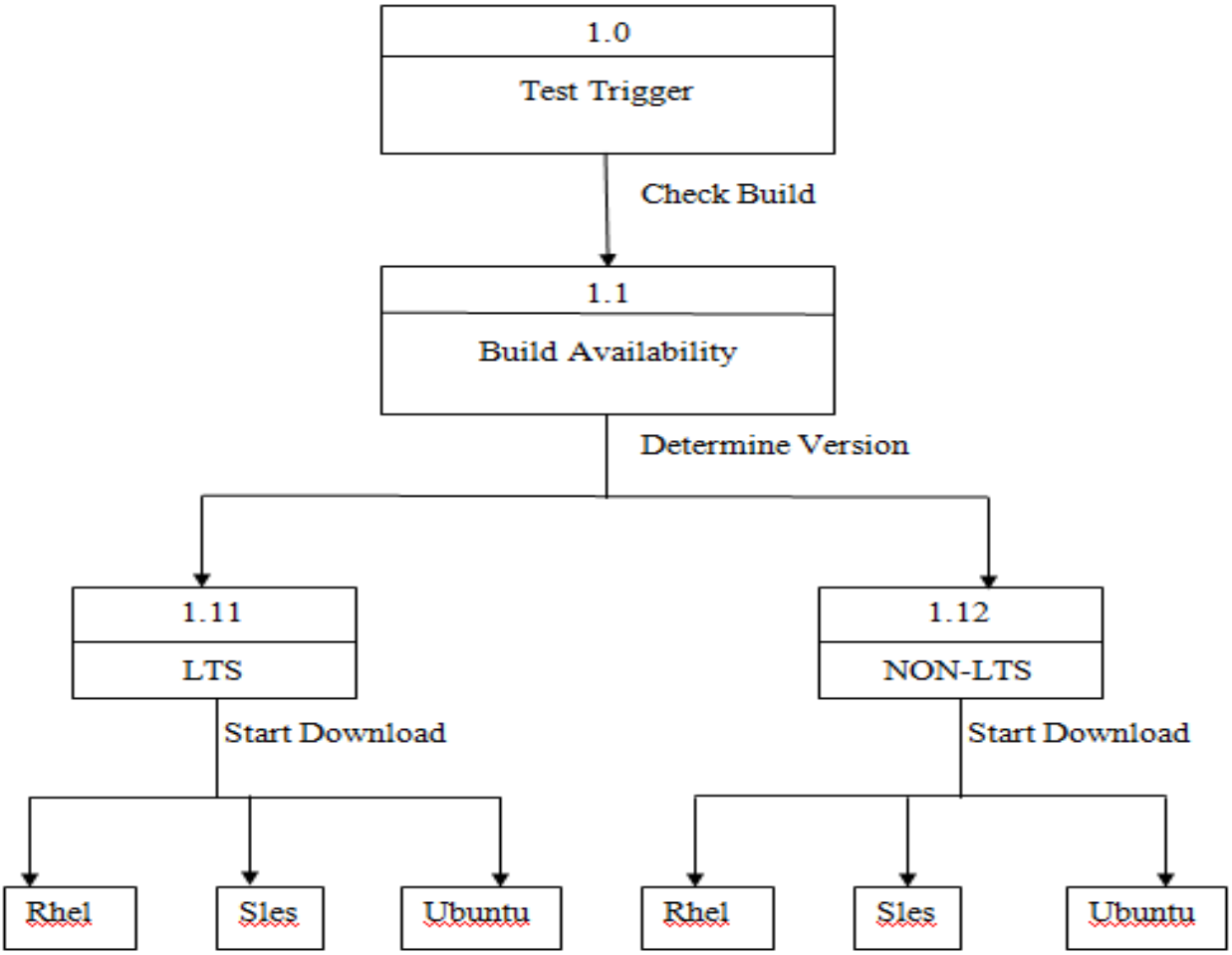


Figure 3: Flowchart for Build Availability

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

IV. TEST PREPARATION

Purpose: The purpose of this module is to check for the various availability of the system for OS installation and generate optimal test combinations to be executed on the system , which will be used to test for the many combinations.
Functionality: It generates many possible test cases with Test cartesianer and next it will finalize optimal test combination with Test Augmenter and find the system for OS installation by SUT Locator
Input: The input to this module is the Linux Release, firmware details and the list of systems that are available for installation.
Output: The output of this module will be a optimal combination of the tests that are to be executed

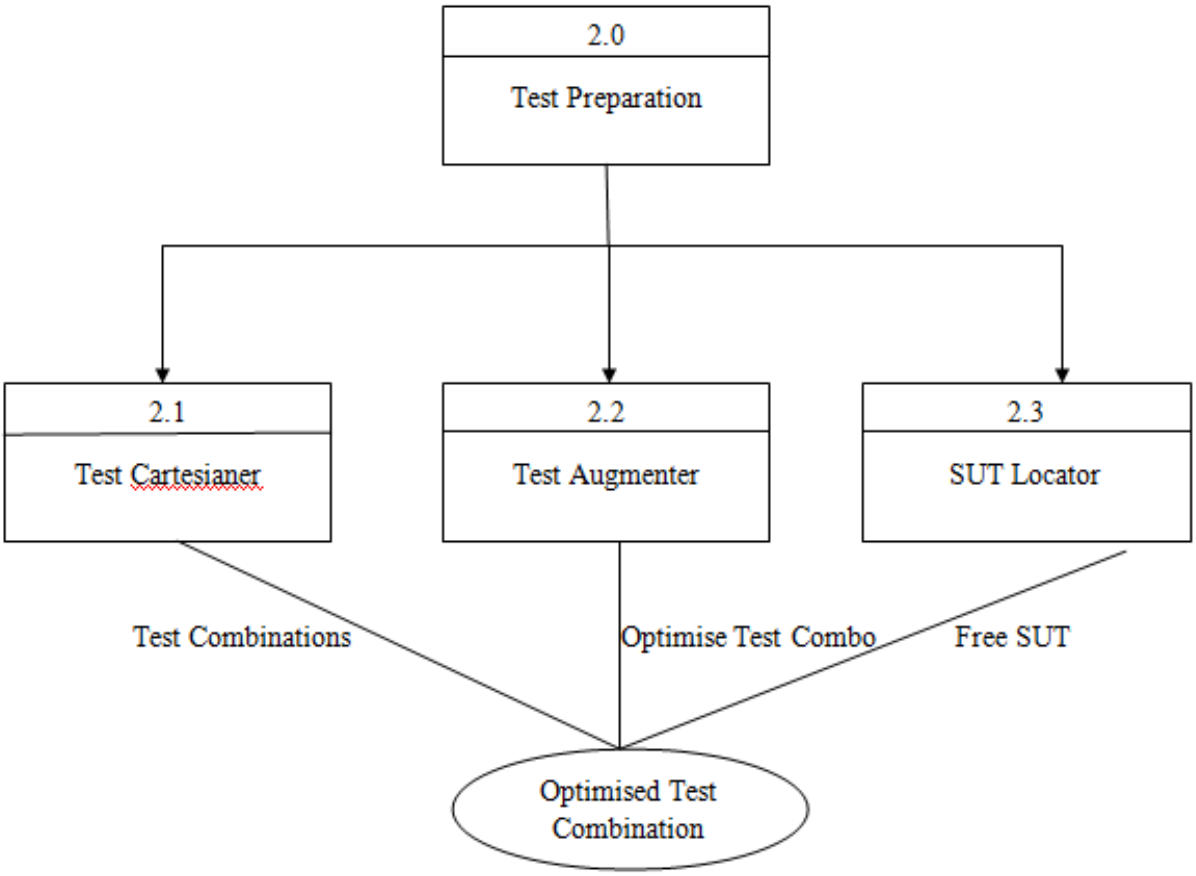


Figure 4 : Flowchart for Test Preparation

V. TEST EXECUTION

Purpose: The main purpose of this is to deploy necessary software and start executing the required tests.
Functionality: It has a module for deploying the required software and get the system for execution and begins execution.
Input: The input for this module is a optimized test combinations.
Output: The output of this module will be the result of execution of the test cases which be even logged to user for analysis.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

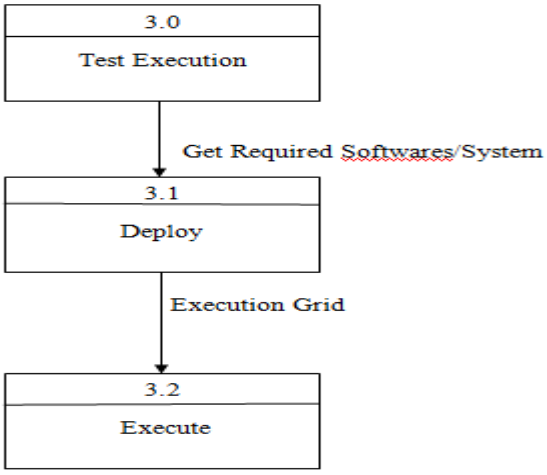


Figure 5 : Flowchart for Test Execution

VI. DASHBOARD

Purpose: The purpose of this module is to display the execution results for the user.
Functionality: It has a module to take user input for the test environment and display the results after execution.
Input: The input for this module is a list of arguments that specify the test environment.
Output: It displays the execution results to the user for analysis which helps user for next iteration.

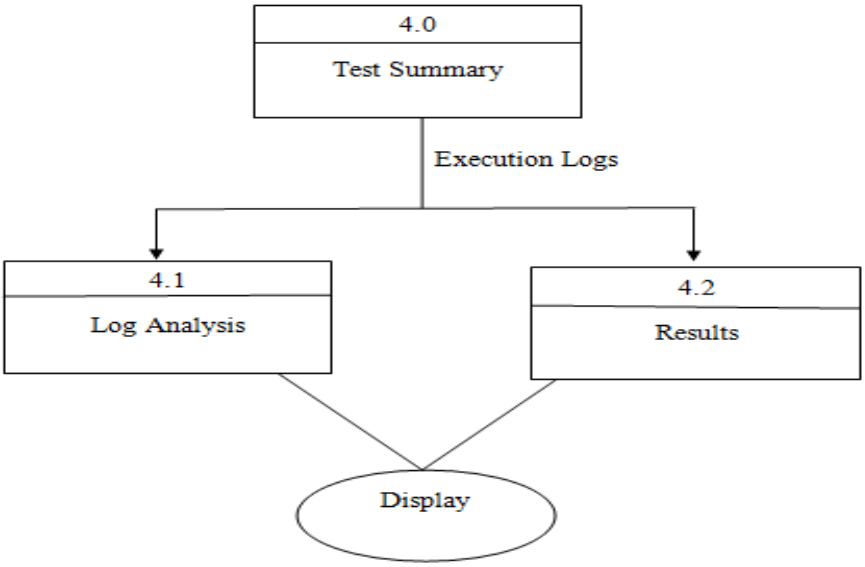


Figure 6 : Flowchart for Test Summary

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

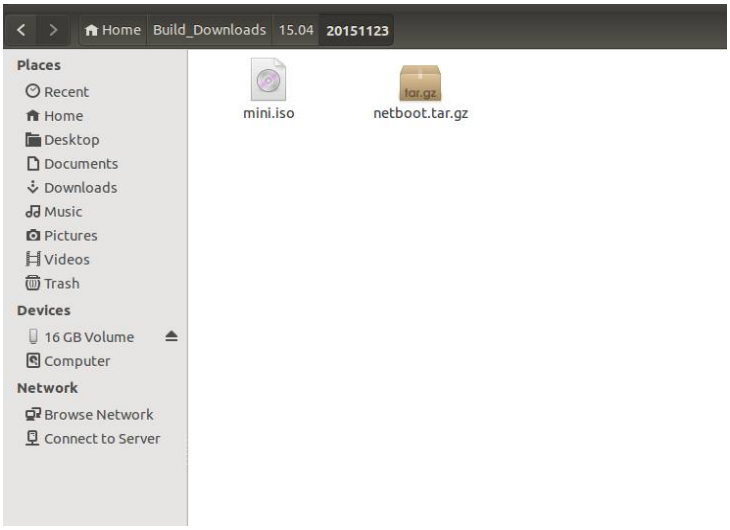
International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

VII. EXPERIMENTAL RESULTS

- Once the code is completed, It is run in such a way that once it starts executing it will run continuously.
- Until and unless the code is stopped manually it can't be stopped. The code should run once in a week automatically.
- Code is written in such a way that it will execute once in a week.
- When there is a new build available in the website, then it will be downloaded into the repository. If not it will not be downloaded.
- The code will execute its other parts only when the new build availability is satisfied or else it will stop.
- Once the condition is passed, it will start downloading the required OS into the respective repository
- screen shot shows the ISO and NETBOOT file being downloaded into the local directory.



SCREENSHOT 1 : FIGURE SHOWING THE DOWNLOADED FILES

VIII. CONCLUSION

The Objective of this project is to develop a end to end continuous regression test framework to perform optimal testing of various combinations of hardware and software. The time for execution of this project is considerably reduced as most of the test cases are being executed in parallel there by having an upper hand on serial execution which takes considerably more time for execution. As most of the things are automated it includes less man power and execution hours there by reducing the overall maintenance cost of the project. The Project is being integrated with Jenkins Framework there by making the best usage of available resources. It is also made sure that each and every output of a particular instruction that is being executed is proper logged on to user logs for further analysis and feedbacks. The Test Framework can also be developed on other platforms such as windows and iOS, to aid optimal testing of various combinations of hardware and software.

ACKNOWLEDGMENT

We are greatly thankful to our family and friends for the continuous support that has provided a healthy environment to drive us to do this project. We also thank the Principal and the Management of NIE for extending support to this work.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

REFERENCES

[1] http://www.guru99.com/automation-testing.html
[2] Martin Fowler, "Refactoring: Improving the Design of Existing Code"
[3] Craig Larman, "Applying UML and Patterns"

BIOGRAPHY

Rakshith KN was born in Mysore,India (1990). He obtained B.E. in 2012. He is presently a student of M.Tech at National Institute of Engineering , Mysore in Computer Network Engineering. His research interests are in the field of Server Machines.

Dr.T.H.Sreenivas is Professor in the Department of Information Science & Engineering. He has received his Ph.D. from IIT, Madras, M.Tech. from IIT, Kanpur and B.E. from University of Mysore. His teaching and research interests are in the field of Operating Systems, Networking, Schedule Optimization and Wireless Sensor Network.