INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 6.078
(Volume 6, Issue 2)
Available online at: www.ijariit.com

Offline examination system using Raspberry Pi

Nikhil Thakur
nikhil.gt45@gmail.com
G. H. Raisoni College of Engineering and Management, Pune, Maharashtra

Sougandh Shah
souganshah.ss@gmail.com
G. H. Raisoni College of Engineering and Management, Pune, Maharashtra

Suraj Yeole
yeolesuraj77@gmail.com
G. H. Raisoni College of Engineering and Management, Pune, Maharashtra

Shivraj Deshmukh
shivrajd9@gmail.com
G. H. Raisoni College of Engineering and Management, Pune, Maharashtra

Sunita Nandgave
sunita.nandgave@raisoni.net
G. H. Raisoni College of Engineering and Management, Pune, Maharashtra

ABSTRACT

India is developing, cities are advancing, but still the villages and remote areas are facing problems. We are still dealing with major challenges pertaining to digital sectors. Even though the internet, penetration has evolved in India, we are still facing continuous network availability issues other discrepancies. This is significantly a high alarming factor for the education sector. To solve this problem, we have come up with a unique robust solution. So, we have developed a Linux based operating system for embedded devices. The operating system is strict as per user requirement for the particular application here it for offline online examination where we break the network into convenient units to overcome the mentioned drawbacks. This mechanism ensures offline conductance of exams in the intranet environment. Further, this will be re-integrated into the consequent network.

Keywords— *Embedded Devices, Operating System, Linux*

1. INTRODUCTION

Embedded Linux is specifically customized for embedded systems Embedded Linux is an emerging technology. Embedded Linux will be applied to more and more devices in the future. Over the past 10 years, embedded Linux has been applied to many devices. From robot to the router, from the TV set to a smartwatch from game box to a car head unit, embedded Linux is anywhere. Especially after the Android system was published in 2006, the application of embedded Linux is booming up. The embedded processors are more and more powerful now. There almost no difference in an application between its embedded Linux version and PC version. So, we have developed a Linux based operating system for embedded devices. The operating system is strict as per user requirements for a particular application. Our operating system is named as CROS. The main feature of CROS is it provides security to the Embedded Devices. Unauthorized human intervention is strictly avoided by CROS.

As we know nowadays, all the exams take place in online mode the major problem is faced by aspirants in rural areas. The trouble lies in the lack of resources available to these students such as inadequate infrastructure, internet connectivity strong mobile network. High costs of data packs make it difficult for rural household to invest their limited saving in these accessories. We have implemented one use case on our operating system, it focuses on internet connectivity issue for online examination in rural areas. We have come up with a unique feature that can help to conduct exams in an offline intranet environment. Exam results can be uploaded to the main server from the local machine insecure way. This will help to manage exams without any need to worry about internet speed. We know that the internet of things (IoT) is projected to soon interconnect tens of billions of new devices, in large part also connect to the internet. The IoT concepts have attracted a lot of attention due to its capability to translate our physical world into a digital cyber world with meaningful information. The IoT devices are smaller in size, sheer in number, contain less memory, use less energy and more computational capabilities. So, if we reduce the size of OS, the remaining memory can be utilized for other tasks which indirectly increases the efficiency of the IoT devices.

Now a day, Linux OS is the choice for almost all new embedded device projects. Linux provides a powerful, flexible kernel and open runtime platform which is being; improved by the Linux community to support the new processor, buses, devices, and protocols. But, the problem here is the process to develop a Linux based operating system for embedded devices is way too difficult and complicated which costs more time and money. So Yocto Project (YP) provides an easier way to create a Custom Linux OS for any device. Yocto Project is an open-source collaboration project that helps to create custom Linux-based systems regardless of the hardware architecture. It provides a set of integrated tools to make working with embedded Linux successful, including tools for automated building and testing, processes for board support and license compliance. ‘We also have the second prototype of the project i.e. Docker. Currently, IT industries are making Docker-compliance applications. So, our OS will be Docker Ready which will serve all types of industry applications. Docker is a tool designed to make it easier

to create, deploy and run applications by using containers. So lastly, our OS will be more efficient and scalable. It will support all Docker-compliance applications.

2. PROBLEM DEFINITION AND OBJECTIVES

(a) To develop a Linux based Operating; System for Embedded Devices.
(b) To develop an Embedded Device to solve the problem of the online examination system in rural areas.

3. LITERATURE SURVEY

In this section, we consider different systems and the technologies that they use.

In [1], In a survey conducted by Oliver Hahm and Emmanuel Baccelli a study comparison between the IoT OSs, their features are done. Studied different kinds of architecture approaches like the Exokernel approach, the monolithic approach, the microkernel approach, the hybrid approach, and its technical properties. The Internet of Things (IoT) is projected to soon interconnect tens of billions of new devices, in large part also connected to the Internet. IoT devices include both high-end devices which can use traditional go-to operating systems (OS) such as Linux, and low-end devices which cannot, due to stringent resource constraints, e.g. very limited memory, computational power, and power supply. However, large-scale IoT software development, deployment, and maintenance require an appropriate OS to build upon. In this paper, they analyze in detail the specific requirements that an OS should satisfy to run on low-end IoT devices and we survey applicable operating systems, focusing on candidates that could become an equivalent of Linux for such devices i.e. a one-size-fits-most, open-source OS for low-end IoT device.

In [2], Emmanuel Baccelli, and Cenk Gundogan Studied different kinds of architecture approaches like Exokernel approach, monolithic approach, microkernel approach, hybrid approach and its technical properties. E. Baccelli introduces RIOT OS, an OS that explicitly considers devices with minimal resources it is a free open source operating system. He states that the internet of things is characterized by heterogeneous devices. These devices ranges from lightweight sensors and are powered by 8-bit microcontrollers to devices equipped with 32-bit processors. The Internet of Things (IoT) is characterized by heterogeneous devices. Neither traditional operating systems (OS) currently running on Internet hosts, nor a typical OS for sensor networks are capable to fulfill the diverse requirements of such a wide range of devices. To leverage the IoT, redundant development should be avoided and maintenance costs should be reduced. RIOT OS allows for standard C and C++ programming, provides multithreading as well as real-time capabilities, and needs only a minimum of 1.5 kb of RAM.

In [3], Arslan Mussadiq, Oliver Hahm and Heejung yu in Survey on Resource Management in IoT OS Study of the open-source OS like Contiki, FreeRtos and tiny OS. Comparison and survey about these OSs related to their process management, memory management, energy management, communication management, and file management. In this paper, an effort is made to provide insight into various proposed approaches in the IoT OS resource management research area. This paper provides the characteristics of different IoT OS protocols, their design strategies, along with their relevant advantages and limitations. The contributions are multi-fold. First, the IoT concept, various standardization efforts, and motivations to study the management of IoT resources through an IoT OS are provided. Second, various previously surveyed papers are discussed. Third, each resource management aspect of Contiki, TinyOS, and FreeRTOS is elucidated. Their resource management mechanisms are classified into various sections, including process management, memory management, energy management, communication management, and file management.

In [4] Mahendra swain and Abhishekh Shrivastava discussed that the Linux kernel customization for ARM11 platform based Raspberry Pi. In this customization of the kernel using different toolchains is done. It will show how to configure and cross-compile Linux kernel for the Raspberry Pi on Ubuntu 12.04. An overall idea of building a Linux kernel for Raspberry Pi using the Yocto Project. The custom Linux kernel synchronizes the time, scheduling, resources allocation and managing of all the hardware peripherals and requires less memory. The Linux kernel and associated open source infrastructure is the core of a new ecosystem for the embedded operating systems, infrastructure, and application prototyping, optimization, and deployment. They have discussed the configuration Linux kernel for advanced ARM processors.so it is necessary to update the old Linux kernel when that becomes not appropriate for interrupt handling, scheduling different tasks, resources allocating, management of on-chip memory, multitasking, and Easy user interfaces. The porting of the Linux kernel on a target platform depends upon a number of factors. It concerns with the Linux kernel configuration and compilation for the Raspberry Pi on Host Ubuntu 12.04. Toolchains are build up around cross compiler and executable files can port in the target platform.

In [5], Weisong shi states that with the proliferation of the Internet of things (IoT) and the burgeoning of 4G/5G network, we have seen the dawning of the IoE (Internet of Everything) era, where there will be a huge volume of data generated by things that are immersed in our daily life, and hundreds of applications will be deployed at the edge to consume these data. Cloud computing as the de facto centralized big data processing platform is not efficient enough to support these applications emerging in the IoE era. Leveraging the power of cloud computing, edge computing has the potential to address the limitation of computing capability, the concerns of a response time requirement, bandwidth cost saving, data safety, and privacy, as well as battery life constraint. “Edge” in edge computing is defined as any computing and network resources along the path between data sources and cloud data centers. Finally, they present several challenges and opportunities in the field of edge computing.

In [6], Mung Chiang writes about fog which is an emergent architecture for computing, storage, control and networking that distributes these services closer to end-users along the Cloud-to-Things continuum. It covers both mobile and wireline scenarios, traverses across hardware and software, resides on network edge but also through access networks and among end-users, includes both data plane special cases like cloudlets and control plane special cases such as crowd-sensing. As an architecture, it supports a growing variety of applications, including those in the Internet of Things (IoT), Fifth Generation (5G) wireless systems, and embedded artificial intelligence (AI). This survey article summarizes the opportunities and challenges of Fog, focusing primarily on the networking context of IoT.

4. SYSTEM DESIGN

Initially, the embedded device is first switched-on and hotspot is created. The users are connected to the hotspot. The Static IP is provided to the user to appear for the exam. 9 The user-provided with static IP Will get the login page. The user will enter the

username and password and will give the examination. As the examination is complete the result is displayed on the screen and stored in the database that is on the embedded device. The embedded device act as a local server as well as stores the data. The data that is the results are sent and stored into the AWS Cloud services. The data that is the results are sent and stored into the AWS Cloud services. The admin can retrieve the results from the main server.

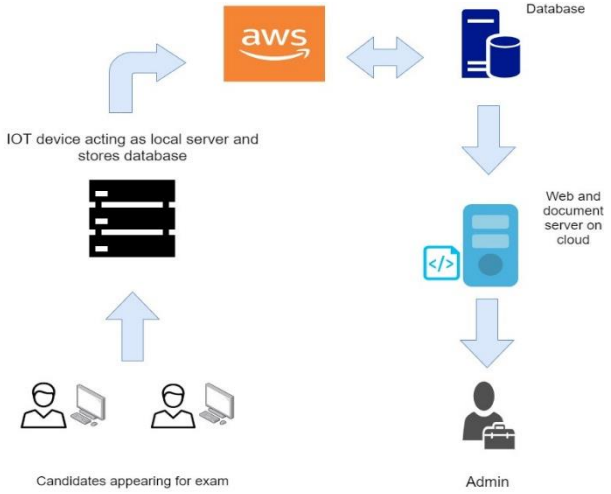

Fig. 1: System design

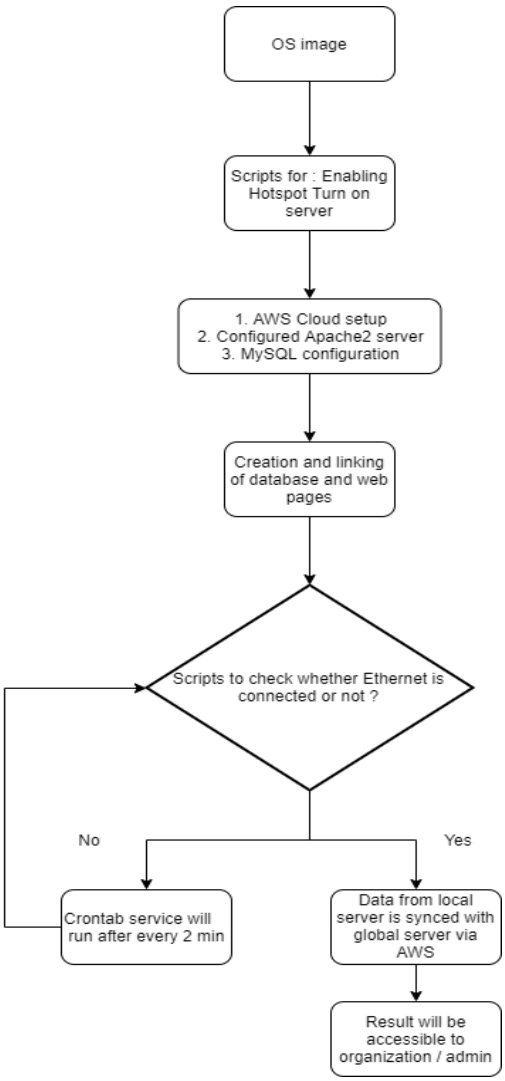

Fig. 2: Flow control

5. RESULT

The system is an embedded Linux product that helps students in giving examinations in an offline environment.

6. CONCLUSION AND FUTURE WORK

6.1 Conclusion

We have developed a Linux based operating system for embedded systems. We have referred to different research papers and compared them different. Operating System, their processing, memory storage, computation power. Different tools and technology are used to develop this Linux based operating system. The OS can be customized as per the user requirement and provide various features. Hence, one of the use-cases is successfully implemented as a closed system for the problem of the examination system in rural areas by customizing our operating system for the application.

6.2 Future Work

- To implement and provide Firmware over the Air service (FOTA).
- To implement the Captive Portal.
- To deploy the operating system on Cheaper Development Boards.
- To implement Remote Proctoring.

7. ACKNOWLEDGMENT

We would like to show our gratitude to our project mentor and teachers from G H Raisoni College of Engineering and Management, Pune for sharing their pearls of wisdom with us during the course of this research

8. REFERENCES

[1] Arslan Mussadiq, Yousaf Bin Zikria, Oliver Hahm, Hccjung yu. Ali Kashif Bashir "Survey on Resource Management in IoT OS" IEEE IOT Paper, February-2018.

[2] Oliver Hahm, Emmanuel Baccelli, Hauke Petersen, and Nicolas Tsiftes" OS for the low-end device in the IoT': A Survey" IEEE [OT Paper, February 2016.

[3] Emmanuel Baccelli, Cenk Gundogan, Oliver Hahm. Peter Kietzmann, Martine S. Lenders, Hauke Petersen, Kaspar Schleiser, Thomas C. Schmidt, and RIOT: Open source OS for Embedded Devices IEEE IOT Journal 2018.

[4] Mahendra Swain, Abhishek Kumar Srivastava Custom Linux Kernel for Raspberry Pi Using Ubuntu I USE Journal of Electronics and Communication Engineering (IOSR-JECE), 2016.

[5] Christopher Hallinan A Book: Embed (led Linux Primer, 2nd Edition a Practical, Real-World Approach Reference Book.

[6] Yocto Project Manual. Available: https: / / www.yoctoproject.org/ docs / 2.4 /ref manual/ref-manual HTML. Yocto Project Manual.

[7] Yocto Manual. Available:https://www.yoctoproject.org/docs/2.4/devmanual/devmanual.html. Yocto Project Manual.

[8] Weisong Shi, Jei Cao Edge computing vision and challenges (March 2018)

[9] Shanzhi Chen, Hui Xu A vision of IoT: Applications, Challenges, and Opportunities (May 2018)

[10] Mung Chiang, Tao Zhang Fog and IoT: An Overview of Research Opportunities. (June 2018)

