International Journal of Software Innovation
Volume 9 • Issue 2 • April-June 2021

Design and Development of Lightweight Operating System Framework for Smart Devices

Jasleen Kaur, Department of Computer Science and Engineering, Indira Gandhi Delhi Technical University for Women, India

S. R. N. Reddy, Department of Computer Science and Engineering, Indira Gandhi Delhi Technical University for Women, India

ABSTRACT

Immeasurable extension of technology for smart devices is perceived in the present scenario, and has resulted in advancements in the growth of such smart devices in diverse domains. An application specific, customized, and lightweight operating system (OS) eventually builds the complete smart device. The user may face difficulties in developing application specific customized OS because of lack of knowledge or resource constraints. The aim of this paper is to assist the users through a generic smart framework that significantly reduces the constraints in the design and development of smart devices and help the users to automatically build an application specific lightweight customized OS. The targeted framework comprises an intelligent graphical user interface that directs the selection of processor and application specific key components. The performance analysis of the framework is done through an experimental test on two different application areas. The findings infer that the size and boot time of an OS build from the framework diminishes by more than 50%.

KEYWORDS

MySmartOS Framework, OS for Smart Devices, OS Framework, Selecting Application Specific Components

1. INTRODUCTION

Smart devices are electronic devices connected to a network that can operate in an interactive and autonomous way (Silverio et al., 2018). In today's era, smart devices are becoming ubiquitous. These devices offer a variety of applications that can meet virtually any need at the touch of a button (Muhlhauser, 2008). With the fast advancement of Internet technologies, smart devices with network capabilities have grown in popularity (Kaur & Reddy, 2017). The Smart device development is inclined to advance the creation of technology-enabled ecosystem which is continuing to evolve. Smart devices contain several basic components, such as sensors, a communication module, multimedia devices and other input / output peripherals (Wipro & UBM Tech, 2013). To develop a smart device specific to an application, the user must select a combination of the basic components needed for a specific use case (Wipro & UBM Tech, 2013). An integral aspect of a smart device is an OS with an application specific functionality for better control between interfaced modules and a better user experience

DOI: 10.4018/IJSI.2021040108

(Muhlhauser, 2008). The operating systems such as Linux and Android are open source (Kute, 2012) and a user can use any open source OS to develop smart devices. A critical design problem with such devices is an OS that should be lightweight, for low end smart devices. Historically, most operating systems are known for their size, boot time and complexity. An OS supports every possible software feature, but most of the applications only need a subset of the functionalities (Kaur & Reddy, 2017).

The Low-end smart device requires a lightweight OS (Hahm et al., 2015), considering their special characteristics and to customize it for better efficiency. Based upon the survey in (Kaur & Reddy, 2017), it has been concluded that users find it challenging to develop an application specific lightweight OS image for different reasons including lack of knowledge or resource constraints. Therefore, an open source framework to automatically generate the application specific lightweight customize OS, is required.

This paper discusses the design & development of a new generic framework for ARM architecture-based processors to meet the user's expectations, so that they can automatically create an application specific and lightweight OS image for a smart device. The targeted system also includes an advanced, user-friendly interface, which recommends the essential key components for the specific application. The essential components such as sensors, communication modules, multimedia modules and other Input/output peripherals are recommended to the users based upon two factors: a) the selected smart device application name; and b) Cost (low or high).

The remaining paper is organized as: the literature survey summary and gaps in the research are defined in section 2. Section 3 construed the methodology that comprises architecture, technology used, brief description of design & development and the implementation of the targeted framework. Section 4 explicates the experimental results with two different application areas. Section 5 summarizes the conclusion followed by the future work.

2. LITERATURE SURVEY

An OS should be customized for the specific application to overcome the problem of memory footprint, redundant code, boot time and complexity of an OS. This section discusses the study of the literature on various existing frameworks. Moreover, the survey gives the conceptual basis and information about what has been done in the field of an OS customization.

(Gauthier et al., 2001) targeted heterogeneous application areas for automatic application SW and proposed automatic generation of application specific OS relying on OS code library with macro language codes. ARM7 processor is targeted with tree-based approach for specific APIs in task scheduler. Architecture analyzer, code selector and code expander were used to generate a processor specific make file to be used in the OS kernel. The proposed method is tested on a VDSL framer and a token ring system.

(Chang & Chang, 2003) proposed a minimal overhead extensible kernel in their framework: OS Portal that follows the client - server model. Client kernel is extensible by downloading of those application specific requirements that are not present in the current version. Server model took care for symbol generation using nm, GNU binary utility. Via symbol table of client version, OS Portal links required module to modify Linux kernel.

(Krause et al., 2005) presented platform independent software development idea for automated code generation for RTOS running on the target processor. 5 ways of refinement are processed for transaction level modelling. The refinement process chooses the target platform and convert the target OS into client specific OS through different level of abstraction.

(Sato et al., 2007) presented a hyperlink-based framework, Hyrax for a phone's user interface customization. They used W3C defined XLink and EFI defined WAP forum for object generation and framework evaluation. Hyrax runtime environment includes a micro browser, builder and runtime environment where menu keys are linked to hypertexts for further navigation between different key options to enhance approachability of user to phone functions.