Intelligent Design for SMS module by Ranked based Pre-emptive Job Scheduling Algorithm

***Sasikumar Gurumurthy*[1], *Kumar Raja*[1], *Chandra*[2], *Sivaprakash*[2]**
Sree Vidyanikethan Engineering College, Tirupati, India
Electronics Engineering, mithrangurugsk@gmail.com

Abstract

In the recent years, mobile phones have become one of the most essential electronic devices in our day to day life. Just like other electronic devices, mobile phones too work on OS. The most commonly used OS are Android mobile OS and Windows mobile OS. Android Mobile OS is an open source platform and was developed by Google based on Linux kernel. Windows Mobile OS was developed by Microsoft Corporation which is a closed source platform and it is deemed as a new generation of the OS, it is definitely an OS that we need to look out for. Windows mobile OS uses Windows CE based architecture at kernel level. Just like its counterparts it provides basic functionalities like process scheduling, file management, memory management, resource allocation etc. Out of this, technique scheduling performs an important role in standard performance of any OS because a super platform is that in which no useful resource struggle arises. on this paper we goal to attention on making use of Ranked primarily based pre-emptive task scheduling set of rules on SMS utility of home windows mobile OS. here person affix high ranked to positive contacts, say often used contacts like mother and father or instructors or friends, in such instances if an SMS from those contacts are acquired the proposed utility could render a pop-up notification displaying this high ranked message by means of interrupting on-going process on the display. In flip the high ranked message might be redirected toward and saved in ranked inbox rather than default in container, so that the user can effortlessly get right of entry to the message as and whilst required.

Index Terms: *Mobile OS, Windows OS, SMS application and Pre-emptive Job Scheduling.*

INTRODUCTION

OS acts as a platform between its person and hardware. An OS is considered to be efficient if its reaction time is much less and throughput rate is high. This can be completed via memory management as well as job scheduling set of rules. In addition a cellular OS is a software program platform on which application for cellular devices can run, cell OS is used to complete particular jobs with a given time closing dates. As the response time is an important factor, using ranked based pre-emptive job scheduling would be an efficient solution. This paper highlights the application of Ranked based pre-emptive job scheduling for the SMS module. SMS] is a service component of phone, web, mobile communication systems that uses standard communication protocols to send and receive information through network.

In recent years SMS generation advanced to carry binary statistics including images, motion pictures, ringtones, commercial enterprise cards, voice messages and so on. So there is a want to increase a way that would be able to distinguish critical SMS and everyday SMS based totally on their assigned ranked. This in turn could help the user to obtain the messages easily. This idea is a logical

extension of Gmail’s ranked inbox where an user can segregate messages based on higher prioritized contacts, as a result of which a user can arrange the messages in case of bulky messages in the inbox. So this proposed system makes use of user's contact list to set the high ranked to the required contacts which makes the application to display the important the messages as a pop up while the user is involved in other jobs.

The paper is divided into 7 sections. The Section I of the paper deals with introduction to the paper. The section II of the paper deals with the explanation of the Windows mobile OS architecture. The section III of the paper deals with the explanation of Ranked based pre-emptive job scheduling technique. The section IV of the paper deals with the explanation of algorithm, flowchart of the proposed approach. The section V offers with the experimental consequences. The segment VI deals with the proposed system benefits over the existing gadget. The phase VII deals with the observations and end of the proposed approach.

WINDOWS MOBILE ARCHITECTURE

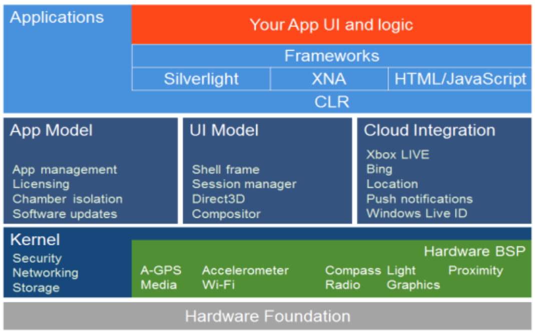


Fig.1. *Windows Mobile Architecture [1] [3].*

Given below are the components present in Windows Mobile Architecture [1] [3]:

Hardware

- Resolution, Camera, Sensors, Capacitive Touch, Multimedia.
- Memory, CPU, GPU, Hardware buttons

Kernel

- Virtual memory, Paging, Security, Networking support.
- Device drivers (written by Microsoft for consistent model).

App Packing Model

- An application is uniquely identifiable, licensable and serviceable.
- Software or application product is packed as XAP file containing compressed assemblies and resources mostly in ZIP file format.

UI Model

- In terms in the UI model an application is user interface components plus logic exposed through one or more pages.
- A page is a single strain made up of UI elements, a session is an ordered workflow of user interactions and in the Windows Phone UI model it can span multiple applications.

Cloud Integration

- Microsoft Services, Existing Web Services or creating your own Web Services.

Application Runtime

- Microsoft Silverlight (Morden event driven application UI Framework).
- XNA (High performance game Framework).
- Both Framework are written as Managed Code thus sharing common set of platform components running in Dot Net Common Language Runtime (. Net CLR)

ANDROID MOBILE ARCHITECTURE

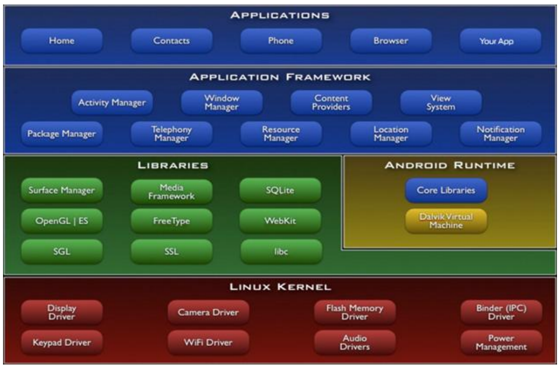


Fig. 2. *Android Mobile Architecture [9].*

Given below are the components present in android mobile architecture:

Application

- It is the top layer in the android mobile architecture where all the applications are stored. Some applications will be preinstalled like Web Browser, Dialler, Contact Manager, SMS Client app.

Application Framework

It contains blocks which can interact with our applications. The replacement and reuse of the components is enabled. Important blocks of the application framework are:

- Activity Manager
- Content Provider
- Telephony Manager
- Location Manager
- Resource Manager

Libraries

Enables the mobile device to handle different types of data. Some of the important libraries are:

- Surface Manager
- Media Framework
- SQLite
- Web kit
- Open GL

Android Runtime

It comprises of Core Java Libraries and Dalvilk VM.

- Dalvik VM is used in android and it is a type of JVM used to run the apps and it is optimized for low memory environments and low processing power.
- Core Java Libraries provides maximum functionalities which are defined in SE libraries.

Kernel

It is the basic layer of the android architecture. The whole android OS is built upon this layer. Linux interacts with all the hardware and contain all the device drivers. Drivers are codes that communicate with the hardware.

PRE- EMPTIVE JOB SCHEDULING

Job scheduling is the vital part in the OS which enables multiprocessing and multitasking. Job scheduling concept explains how a common CPU can be shared among different processes. This duty is carried out by scheduler and dispatcher. A completely fair scheduler is the scheduler where the processes are scheduled with in a constant amount of time in spite of many processes in contention [6].

The pre-emptive job scheduling technique involves disruption of lower ranked jobs when the queue consists of higher ranked jobs [6]. This pre-emptive job scheduling is used in mobile OS because of high response time, and high turnaround time and medium CPU utilization. For some specific jobs to occur the mobile OS have to meet some specific time limits.

EXPERIMENTAL STEPS

This segment details with the stairs carried out to our proposed Ranked primarily based pre-emptive process scheduling technique. In our proposed technique we use List Action as hobby, and List Outlook as object, and Broadcast Receiver as magnificence.

The steps are listed below [7] [10]:

- Firstly, the SMS application would sign in to a category named List Action which binds to the facts source and then shows the list of the gadgets. Whilst an item is selected with the aid of a user, List Action class hosts a List Outlook object restricted to one of a kind information sources.
- Now the consumer can set the ranked “excessive” for a few contacts as per his will whereas the other contacts might stay default. This may be finished with the aid of an interface proven via the software.
- Whilst a message from a excessive ranked contact is conveyed to the cell

even as the user is busy with some other sports at the mobile, then the message via the high ranked touch might be pop-upped on the active screen.

- Simultaneously this high ranked message can be redirected and stored inside the Ranked inbox and will also be saved inside the default inbox.
- when an SMS is conveyed with the aid of the default contacts there may be no disturbance on active(on- going) screen for the person busy with some other jobs, robotically the SMS could be redirected and stored within the default inbox.
- The SMS services information can be determined by the methods in the BroadacastReciever class.

Algorithm

1. Begin the messaging application.
2. Register the delivered SMS.
3. BroadacastReciever receives the SMS.
4. Check the rank of the Contact number.

IF the rank is set to HIGH then

Pop-up the ranked SMS on the active screen and redirect and store it in Ranked inbox as well as default inbox.

ELSE

Store the SMS into the default inbox.

- STOP

Proposed Technique Flowchart

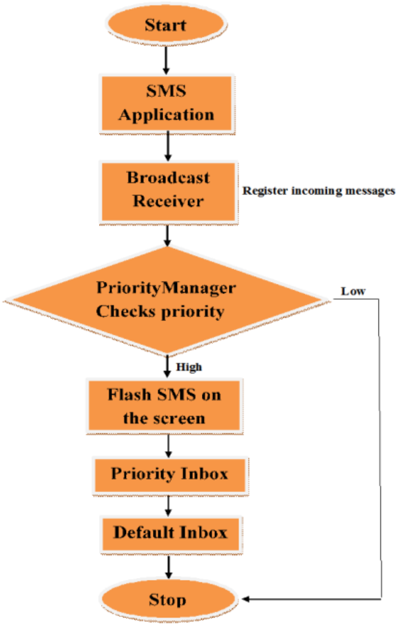
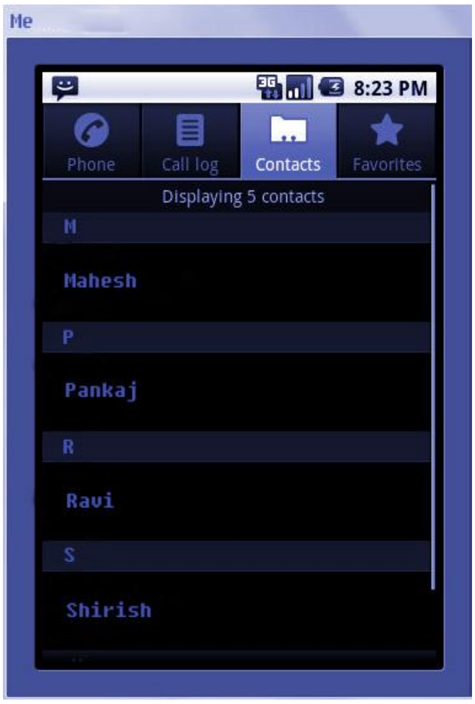


***Fig.3.** Flowchart for the proposed technique [1] [5].*

EXPEMENTAL RESULTS

The Experimental Results are as follows:



Displaying 5 contacts

M

Mahesh

P

Pankaj

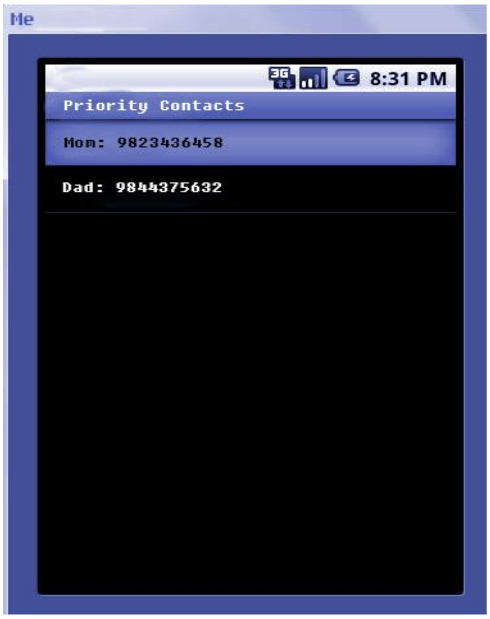
R

Ravi

S

Shirish

***Fig 4:** Contact List*



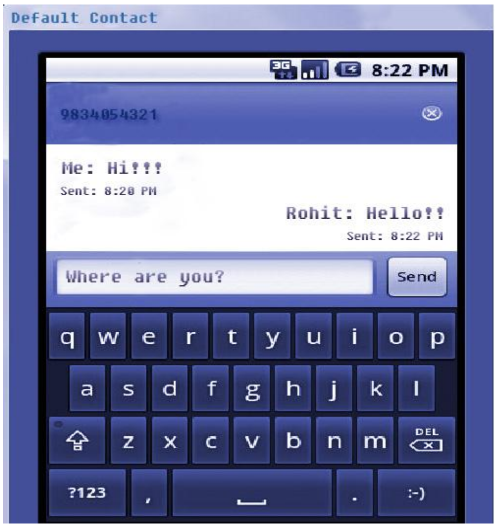
Me

Priority Contacts

Mom: 9823436458

Dad: 9844375632

Fig 5: *Prioritized Contact List*



Default Contact

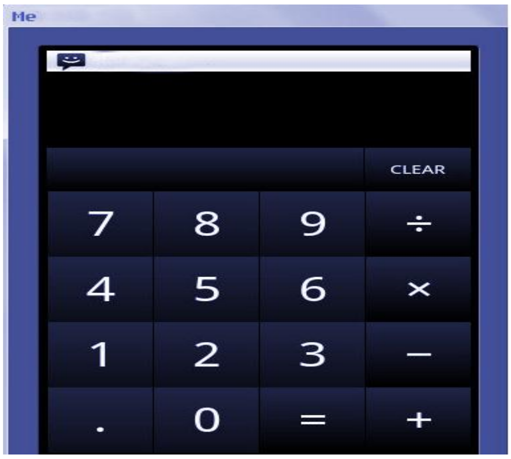
9834054321

Me: Hi!!!
Sent: 8:20 PM

Rohit: Hello!!
Sent: 8:22 PM

Where are you?

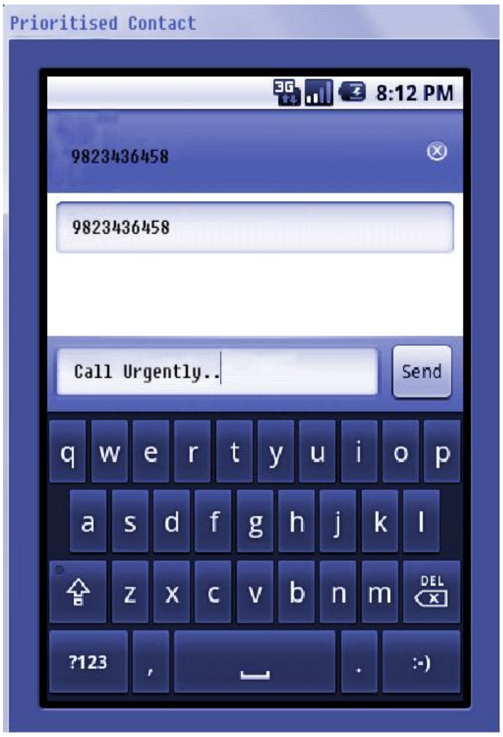
Fig 6: *Default contact typing SMS to be delivered*



Me

Fig 7: *SMS delivered but no interruption*

on the ongoing job and SMS redirected to inbox



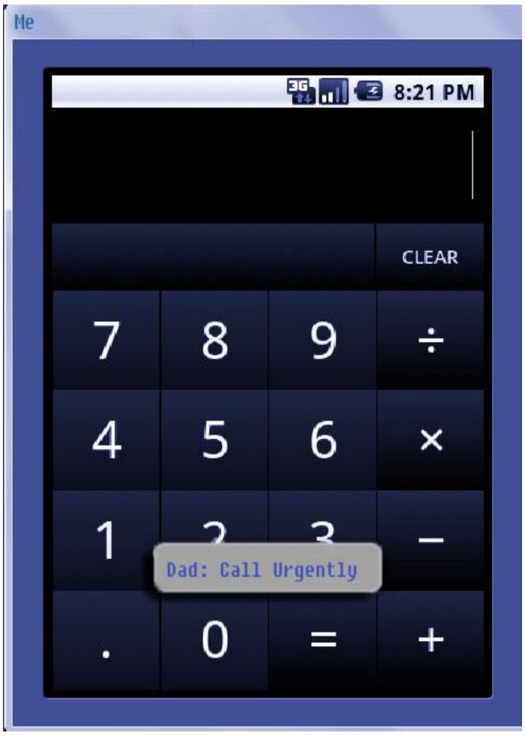
Prioritised Contact

9823436458

9823436458

Call Urgently..

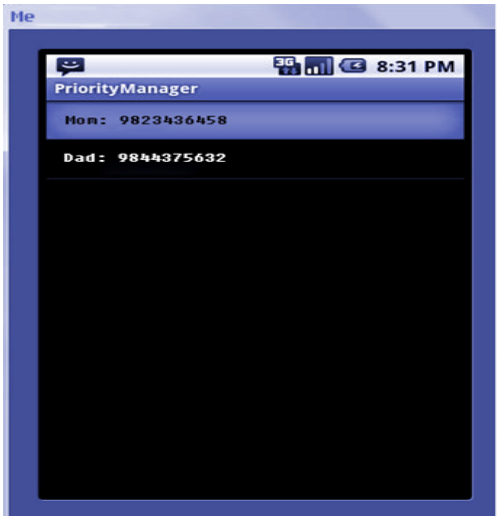
Fig 8: *Prioritized contact typing SMS to be delivered*



Me

Dad: Call Urgently

Fig 9: *SMS delivered interrupting ongoing job and redirected to ranked inbox and default inbox.*

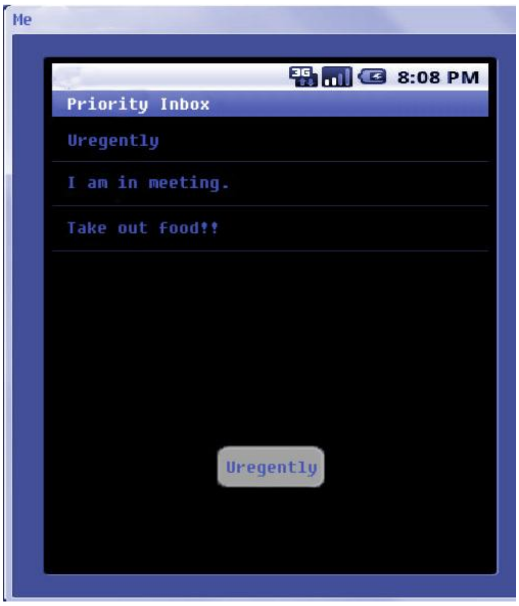


PriorityManager

Mom: 9823436458

Dad: 9844375632

Fig 10: *Selecting the prioritized contact to read the SMS in the ranked inbox.*



Priority Inbox

Uregently

I am in meeting.

Take out food!!

Uregently

Fig 11. *Selected SMS from the ranked inbox flashing on the screen.*

LITERATURE REVIEW

In trendy we stumble upon a certain state of affairs wherein one's inbox gets overwhelmingly crowded with massive variety of messages, following which the user unearths it extraordinarily difficult to type out the vital messages. As an end result and green set of rules is want to separate critical messages from the regular messages. As there are several scheduling techniques available for the OS, but none of them are used to rank the SMS application. Hence, Pre-emptive based job scheduling algorithm would be an efficient way to work around this problem faced daily by the user.

The idea could be borrowed from Gmail's ranked inbox. In Gmail the feature of the ranked inbox ranks the e-mails on its arrival. A user statistical model is prepared in order to predict this frequency. Also this frequency is updated as frequently as possible [11].

Similarly, this concept can be implemented in the **proposed system**. In this algorithm a low ranked job can be masked when a high ranked job arrives (i.e.) If a user receives SMS from higher ranked contacts that he/she is allowed to assign to the required contacts such as parents or friends contact numbers, users ongoing job is paused, the application would alert the user by popping up the SMS on the active screen and redirects the higher ranked SMS to the ranked inbox. This in turn minimizes CPU utilization, increasing turnaround time and response time. This particular gain helps a lot as a mobile OS are needed to do specific jobs in limited time frame [6].

OBSERVATION AND CONCLUSION

It is rare that we think of ranking the contacts as 'High' and keeping other contacts as 'Default' resulting in better utilization of the contacts present in the mobile phone. This same application is implemented in the proposed paper. In the proposed paper we are using the SMS application which is used in our daily life. The SMS application is used in our proposed paper because the SMS application is most widely used after the call application. In the proposed paper SMS documents makes use of the proposed technique for home windows cell OS and Android cell OS. In destiny the proposed technique of high ranked contacts and ranked primarily based pre-emptive job scheduling may be used for the decision utility. The proposed approach may be established on different cellular systems like Symbian, Bada and so on.

REFERENCES

1. Windows Mobile Infrastructure Components Windows Mobile 5.0 and 6 – powered Devices Architecture. White Paper published: 2007
2. “Mobile Application Architecture Guide,” Patterns & Particles by J.D. Meier, Alex Homer, David Hill, Jason Taylor, Prashant Bansode, Lonnie Wall, Rob Boucher Jr, Akshay Bogawat.
3. “Windows Phone: WP7 Architecture Overview,” Amar Nityananda, Partner Technical Consultant, Gowtham Prasad K N, Developer Evangelist.
4. “Comparing architecture of mobile applications,” Krešimir Fertalj, Faculty of Electrical Engineering and Computing, University of Zagreb Unska 3, HR – 10000 Zagreb, Croatia, Marko Horvat, Croatian Railway Ltd. Braimirova 9a, HR-10000 Zagreb, Croatia.
5. “The Learning Behind Gmail Ranked Inbox”, Douglas Aberdeen, Ondrej Pacovsky, Andrew Slater, Google Inc. Zurich, Switzerland.
6. MA Wei-feng, WANG lia-hai, “Anaysis of the Linux 2.6 kernel scheduler”, International Conference on computer design and Application (ICCDA), 2010.
7. XU Long, et.al, “An Improved Algorithm to Optimize the Access Time of Data Broadcast in Mobile Computing Environments”, Second International Workshop on Education Technology and computer science, 2010
8. Sangchul Lee, Jae Wook Jeon, “Evaluating Performance of Android Platform Using Native C for Embedded Systems”, International Conference on control, Automation and Systems, 2010
9. Nasr Addin Al-maweri, et,al, “Runtime CPU Scheduler Customization Framework for a flexible mobile OS”, Student Conference on Research and Development (SCOReD), 2009.
10. Li Lo, et.al, “A modified Interactive Oriented Scheduler for GUI-based Embedded Systems”, 8th IEEE International Conference on Computer and Information Technology, 2009.
11. C.S. Wong, et.al, Fairness and Interactive“.