

Design a Mini-Operating System for Mobile Phone

Dhuha Albazaz

Computer Sciences Department, University of Mosul, Iraq

Abstract: Due to the development witnessed in the field mobile phone and the development of their operating systems added to the increase in number of users and many businesses that have relied on them, a large number of programmers have started to develop special operating systems for these phones and building applications that meet the demands of the users and facilitate a great number of businesses. The proposed work is designing a mini-operating system for managing some special features of the mobile phone. The operating system designed in this work is based on multitasking –multithreading with mixing of preemptive and cooperative mode. The functions and features chose where those related to message management for sending and receiving SMS. Photo-album application also is chooses for managing and displaying images with different formats stored in mobile memory, and contact application for displaying names and phone numbers. This system can be considered a starting point to establish an integrated operating system for mobile phones. As the internal memory for the mobile phone is small, low output size language J2ME has been used for programming this operating system. J2ME relays on virtual machines which is an implementation of KVM in its operating. This language is characterized by its multiple channels and it is considered appropriate for all low memory sets.

Keywords: Mobile, smartphone, operating system, multithreading, pre-emptive.

Received August 18, 2009; accepted March 9, 2010

1. Introduction

The mobile phone or cellular phone is one from of telecommunication that is based on wireless connection via a network of broadcast towers distributed within a certain area. The development of mobile phones led them to be more than merely voice communication means [15]. But have started to be used as small computer devices to receive and send e-mails, appointment, surf the internet. Also, the most advanced of them are capable of taking photos which are compared to high resolution cameras. Mobile phones have also become means for promotion and advertisement and as the competition of service provider for mobile phones increase, the price of communication using these phones has become affordable for a wide category of people [16]. Thus, the number of mobile phone users in the world in general and Arab world in special is steadily increasing leading the mobile phone to replace the traditional wire phone in households.

The history of mobile phones goes back to 1947 when Lost Technologies Co. started its experiments in it New Jersey plant. However, the first who accomplished the mobile phone was Martin Cooper a researcher from Motorola in Chicago who made his first wireless call on April 3rd 1973.

The mobile phone is a sending and receiving circuit using frequencies across land based stations while other are completely performed by satellite . The principle is similar to radio communication but differs in its frequencies that are powerful and can reach to 20 MHZ of sending / receiving per second [1]. The

communication can take place using an integrated circuit, the mobile set, main switch at the service company and the SIM CARD. The SIM Card is a small circuit that has a small memory unit, a processing unit that stores the user's data which is used to communicate with others. The mobile phone consists of (receiving circuit-a sending circuit-a central processing unit-a Ram and a flash memory) [7, 13].

The objective of this proposed system is keep in pace with the development in the field of programming embedded programs and acquire the requisite knowledge for these kinds of operating and management systems especially when developing special applications.

1.1. Related Works

Mobiles have become popular among users for their various features especially intelligent mobiles that can access the internet and e-mail. But the unknown soldier that works behind the curtains and securing all the elements is the operating system. The operating system defines the features, performance, security level and the active programming interfaces API. Thus the technical users of mobile relay on the operating systems when they make their purchase decision instead of relaying of the feature of the handset only because a good operating system can operate even on modest handsets which cannot be achieved with other operating systems besides performing the usual tasks of multimedia presentation and SMS which are not a measure of the devices quality [12]. I started by taking

a look at the operating systems for mobile devices, and more specifically for Smartphone. The most common operating systems used in Smartphone are Linux, Windows Mobile from Microsoft, Symbian, RIM BlackBerry, Android, and Mac OSX.

- *Symbian*: Is an operating system optimized for mobile terminals and provides a reliable environment as it has been designed so that user data is never lost and the device running the operating system will never have to be rebooted. Approximately 72% of mobile devices use Symbian and major mobile companies adopt this operating system in their manufacturing including Nokia and Sony Erickson. The reason is the wide choices for the programmer because the system is developed using C++ with plug INS that facilitate its easy use and added features. The system allows the administrators to access remote to the user's mobile to add additional programs and features or diagnose a problem. However, this feature has its disadvantages as viruses can access the mobiles and send information or copy themselves in nearby devices through the Blue tooth technology. This leads the mobile manufacturers to continually update their devices with security programs which are added costs paid solely by the user increasing the price of the set [3].
- *Linux*: Is characterized by its support by a large community of programmers and ready to develop all its programs making Samsung, Motorola, Panasonic and NEC ... etc adept it in their intelligent phone raising its popularity to 13 %. The main problem in this system is the programs in one version do not work in other versions in general because of the unified codes. But some interesting parties are working towards solving this problem [6].
- *Mac OS X*: Apple has shaken the standards of mobile markets and raised the expected standards of the operating systems by introducing the Iphone that works of Mac OS X [5]. The system can receive instruction by touch screen awhile analyzing and interacting with them. The company has introduced a number of programs including Safari which is capable of displaying difficult pages besides multimedia ... etc. Mac OS X is less popular with developers in general because of the non standard Java runtime environment [6].
- *Blackberry*: RIM provides a group of excellent programs for Blackberry that amount to 1500 program and thousands of others through various companies including the programs for photo share map visualization, multi-language dictionary, etc. The Blackberry program is written with Java language which means that the programs can work by many versions and the system can distinguish between secure and suspicious programs allowing only the first to access the users data (names,

addresses, documents and photos) while automatically supporting Email delivery and include high security levels based on advanced encryption for information security [6].

- *Windows Mobile*: The windows mobile system has received great popularity between the programmers and users because it is an easy to use and develop through Dot.net studio and which is compatible with a large number of programs on desk top PC or servers (email computers). Its popularity is 6% and the system comes in two versions: the first is for intelligent phones that are appropriate with wireless Email, calendar and voicemail. While the second is appropriate for pocket Pc by providing smaller versions of office software (Word, Excel, PowerPoint and Outlook Express). There is no multithreading mechanism in this operating system [6].
- *Android*: Is an open mobile operating system for all phones and devices including personal computers. It is a multiple layer OS that is based off of the Linux architecture which makes is super light for mobile phones and completely open. Android includes many layers on top of the Linux kernel, but the main features of the OS is its user interface and its applications [2].

1.2. Contributions

This proposed mini operating system for mobile phone presents some contributions like:

1. Expounding a new type of operating system which is a mini operating system for mobile phones.
2. This proposed mini operating system is very reliable system. Support a real time nature by manages the execution of time-critical tasks in its real-time. Giving high priority to these tasks and guarantee meeting their deadline.
3. Mixing of two modes pre-emptive and co-operative mode for the implementation and scheduling of threads in this design. For the time-critical tasks I use co-operative mode. While for the other tasks I use pre-emptive mode.
4. Using the multitasking with multithreading in this operating system design help to not require extensive system memory management.
5. The strategy in this design is based on constructing separate threads for the user input and the other lengthy operations. This will help to user input thread from being waiting till the lengthy operations threads over finish.

2. Operating Systems for Smartphone

Series 60 platform is targeted at Smartphone. A device having the size and form factor of a normal phone, while providing, a graphics-capable color screen, value-adding applications such as messaging tools

(e.g. e-mail, advanced calendar, and contacts book) and the ability to install new applications is categorized as a Smartphone [3, 13].

In addition to features easily noticed by the user, there are other types of requirements for Smartphone. As they are embedded systems (like all mobile phones), behavior accepted for typical personal computers cannot be tolerated [13]. The operating system of the Smartphone needs to be very reliable and stable – system crashing and rebooting are most undesired features. It must also fit into a very small amount of memory and use the resources sparingly. A real-time nature with predetermined response times is required of the parts that deal with the cellular networks. In addition it is preferred that the entire system provides fast response times [14]. The overall quality of the operating system and the software platform of the Smartphone need to be very good, as the cost of replacing the software with a new version is high [3]. A Smartphone operating system must be robust and support design principles that allow other software to be reliable. Robustness of the operating system is one of the key criteria to be considered when selecting the platform for Smartphone.

The operating system of a Smartphone is the most critical software component as it depicts the nature of software development and operating principles [9]. The most important requirements are multi-tasking (with multi-threading), real-time operation of the cellular software, effective power management, small size of the operating system itself. As well as the applications built on it, ease of developing new functionality, reusability, modularity, connectivity (i.e. interoperation with other devices and external data storage), and robustness [3].

The operating system defines the features, performance, security level and the active programming interfaces API [18]. Thus, the technical users of mobile relay on the operating systems when they make their purchase decision instead of relaying of the feature of the handset only. This is because a good operating system can operate even on modest handsets which cannot be achieved with other operating systems besides performing the usual tasks of multimedia presentation and SMS which are not a measure of the devices quality [3].

3. Multitasking vs. Multithreading

Multitasking is the ability of an operating system to execute more than one program simultaneously. Though we say so but in reality no two programs on a single processor machine can be executed at the same time. The CPU switches from one program to the next so quickly that appears as if all of the programs are executing at the same time. Multithreading is the ability of an operating system to execute the different parts of the program, called threads, simultaneously.

The program has to be designed well so that the different threads do not interfere with each other. This concept helps to create scalable applications because you can add threads as and when needed. Individual programs are all isolated from each other in terms of their memory and data, but individual threads are not as they all share the same memory and data variables. Hence, implementing multitasking is relatively easier in an operating system than implementing multithreading [8, 18].

All processes have distinct address spaces, open files and other system resources. This enables each process to run with little or no effect on other processes running at the same time. However, this separation of resources has a run-time cost:

- Each process consumes system memory for its address space.
- The operating system must update a lot of internal information whenever it switches execution context from one process to another.

Because threads run within a single process, all of the threads associated with a process share the same data and code address space, the same open files and most other resources. Each thread also has its own register set and stack space. Creation of a thread, therefore, does not require extensive system memory [10]. When the operating system finds it has to switch execution from one thread to another within the same process. So the operating system need in general only switch CPU ownership and register sets. This is the reason that threads are often called lightweight processes. The problems with threads are that switching between them costs a lot of resources and the amount of resources on handheld systems are very limited [11]. Threads are also called lightweight processes that appear to run in parallel with the main program. They are called lightweight because they run within the context of the full-blown program taking advantage of the resources allocated for the program [18]. On a single processor system, threads can be run either in a preemptive mode or in a cooperative mode.

4. Pre-emptive Mode

In the pre-emptive mode, the operating system distributes the processor time between the threads and decides which thread should run next once the currently active thread has completed its time-share on the processor [4]. Hence the system interrupts the threads at regular intervals to give chance to the next one waiting in the queue. So, no thread can monopolize the CPU at any given point of time. The amount of time given to each thread to run depends on the processor and the operating system. The processor time given to each thread is so small that it gives the impression that a number of threads are running simultaneously. But, in reality, the system runs one

thread for a couple of milliseconds, and then switches to the other and so on. It keeps a count of all the threads and cycles through them giving each of them a small amount of the CPU time. The switching between threads is so fast that it appears as if all the threads are running simultaneously.

The switching means that the processor stores the state of the outgoing thread (it does so by noting the current processor register values and the last instruction-set the thread was about to perform). Then restores the state of the incoming thread (again by restoring its processor register values and picking the last instruction-set where it had left itself) and then runs it. But this style has its own flaws. One thread can interrupt another at any given time [10, 18].

5. Cooperative Mode

In cooperative mode, each thread can control the CPU for as long as it needs it. In this implementation, one thread can starve all the others for processor time if it so chooses. However, if a thread is not using the processor it can allow another thread to use it temporarily. Running threads can only give up control either if a thread calls a yield function or if the thread does something that would cause it to block, such as perform I/O [11].

6. General Strategy of the Proposed Mini Mobile Operating System

This proposed mini mobile operating system designed in this work is based on multitasking-multithreading. Multitasking is used with the three main applications messages, contact, and the Photo-album. While for designing each application I employed multithreading.

Tasks and threads can be of different priorities Tables 1 and 2. In order to implement multithreading I use Pause and Sleep thread functions. By the sleep function the priority is determined. The pause function in this work is used to implement pre-emption (suspend the current executing thread and transfer the execution control to high priority thread). The priority of sendsms is higher than the priority of image display or name display. Thus when a streamed image is now displaying and a message is received, the scheduler suspends the image display thread. It will do that by executing the pause function and activate the receive message thread. By the completion of receive message thread the display thread resumes commencing from the stop point.

If we need to draw a graphic or do a big mathematical computation and at the same time want to get user input. We first keep all the individual tasks (like drawing an image, or doing computation or asking for user-input) in separate threads. Then we allocate a higher priority to the thread, which is expecting user-input so that its responsiveness is high,

and the thread, which is drawing the graphic or doing the computation, at a lower priority. So that the entire CPU time is not bogged down by these tasks.

Based on the user-input the program has to do some processing. If the processing is long then it may take some time to complete and the user is un-necessarily made to wait till it is over. In such cases, I keep separate threads, one to read user input and the other to handle any lengthy operations based on the input. I think this will make the program more responsive. This also will give the user the flexibility to cancel the operation at any point of the running of the thread. Hence, applications that use user-input should always have one thread to handle input, which will keep the user interface active at all times, and let the processor-intensive tasks execute on separate threads.

Always a thread, which manages time-critical tasks assigned a high priority and the others a low priority. Like the thread listening for calling requests should always remain responsive and hence allotted high priority. We define five priority levels values for threads in this work (PriorityHigh = 1,.... PriorityLow = 5). Threads are scheduled based on the thread priorities or round-robin if the threads have the same priority. The scheduling in this work is preemptive, which means that a higher priority thread may interrupt a thread having a lower priority.

Table 1. Main tasks priorities.

Application	Class Name	Priority
Messages	SMS-Receive	1
	SMS-Send	2
Contact	Loading screen	3
Photoalbum	Photoalbum	4

Table 2. Functions priorities.

Application	Class Name	Function	Priority
Messages	SMS-Receive	StartApp	1
	SMS-send	ComandAction	2
Contact	Loading screen	Loading Screen	3
Photoalbum	Photoalbum	Photoalbum	4
	PhotoFrame	Key Pressed	5

7. Operating System Applications

7.1. Message Application

This application is responsible for sending and receiving messages. Three classes have been built for this application.

7.1.1. First Class is SMSSend

This class will send text via an SMS message connection. It has no runnable characteristic and extended to MIDlet. This extension provides the main function StartApp which represents active state, PauseApp which stops the execution DestroyApp that destroy the application. The steps for this class are with the following algorithm:

Class Smssend Algorithm

1. Define smssend class as a thread with (MIDlet extends, and command listener)
2. Set priority = 2
3. Implement command-action function as a thread
4. If command = "Exit" then //Check the command
 Call Destroy function
 End
 Else If command = ok then
 Call Prompt and send function
 End
 Else stay in step 4

Function Prompt and Send: This function will get the address and check the validity of the phone number, the steps as follows:

1. Get the address from the destination address box.
2. Call function valid phone number //check the validity of the phone number
3. If valid phone number then
 Call function get destination address
 Else display "invalid phone number"

7.1.2. The Second Class is SMSSender

This class prompts for text to send it via an SMS message connection. Runnable class has been defined to use thread sleep function in order to determine thread priority to implement multi-threading characteristic in this design. I use command listener to provide needed command for exiting or ok. The steps that illustrate the operation of this class are explained in the following algorithm:

Class Smssender Algorithm

1. Initialize java library
2. Define smssender class with (Runnable characteristic, and command listener)
3. Get the destination address
4. Call command-action function //analyze the command from user interface
5. If command = back then
 back to display previous screen
 Else
 Try to open the connector
 If connector != null then
 Send the message
 Print "message is delivered"
 Display sending screen
 Close the connector
 Print the connector is closed
 End

7.1.3. The Third Class is SMSReceive

This class receives a text from SMS message connection and displays it. This class has Runnable characteristic with command listener to supply wanted command, It also extended with MIDlet extension to provide main function (destroy App, Pause App and start App). The algorithm is followed:

SMSReceive Class Algorithm

1. Initialize java library
2. Define SMSReceive class as thread with (MIDlet, Command Listener, and Runnable Characteristics)
3. Set priority = 1
4. Define and start function notify incoming message
5. Call function startApp defined as thread
6. Call function Run
7. Call function command action //check command
8. If command = exit then
 Call Destroy function
 End
 Else If command = reply then //reply to message
 Remove leading "sms://" //display the destination address
 Call get destination address // get destination address from sender class
 End
 Else stay in step 8 to get a command

Function Start Application: This function start to create the thread to do the message connection receive text. The algorithm is followed:

Start App. Algorithm

1. Read the sms sender connection (sms + port)
2. If sms sender connection = null then
 Try to open the connector
 Else
 Initialize the text
3. If receiver connection != null || length of connector != 0 then
 Stay in this step
 Else
4. Set alert "waiting message on port"
5. Create new thread with value of sleep
6. Display the resume screen
7. End

7.2. Photo Album Application

This application is capable of displaying photos that are stored in the phone memory. Three classes have been built for this application.

7.2.1. The First Class is Photo Album

This class provides the commands and screens that implement a simple photograph and animation album. It configured the images and animations to be displayed in the descriptor file. This configuration is provided with options to change the speed of display and the frame style. Algorithm below illustrates the work of this class:

Photo Album Class Algorithm

1. Initialize java library
2. Define photo album class as thread with (MIDlet, command listener, & Runnable)
3. If size of list = 0 then Display alert
End
Else
4. Load initial images
5. Call command action function
6. If command = exit then
Call destroy function
End
Else
7. If command = select then
Call function run to display the image
If command != option then
If command = back then go to step 4
Else go to step 6
Else
Call function item state changed // to change the option
If command != ok then go to step 6
Else
Save option
Go to step 6

7.2.2. Photoframe Class

This class provides the picture frame and drives the animation of the frames and the picture. It handles the starting and stopping of the animation. This class implemented as thread to control the timing and receive requests for repainted periodically. It also controls the border style and animation speed.

7.2.3. The Third Class is Test Chart

It is a quick sample of graphics which generates a series of images that can be used as a sample animation. The animation consists of bar chart. The charts are resized to the desired size so they will look good on devices with various screen sizes. The steps are:

1. Initialize java library
2. Define class test Chart
3. Define constructor that initialize height and width requested by image
4. Define function generate image frame
5. Define function draw frame

7.3. Contact Application

This application will display the name and the phone number storing in mobile phone. The details of the work illustrated in the following algorithm:

Contact Application Algorithm

1. Initialize java library
2. Define loading screen class as a thread with (contactlist extension)
3. Set priority = 3
4. Create main form of program
5. Define string with value of names(contact) and its information
6. Determine the key function "Up, down, Left, Right" // perform an operation
7. Set the skin of the form "color, moving, type, animation"
8. End

8. Implementation

This section represents an illustration of the system's execution and presents the system interfaces and the method of executing each program. The system consists of a group of programs represent the applications selected for the cellular phone (message manager, contact, and photo album), as shown in Figure 1.

8.1. Message Manager Program

This program manages the sending and receiving of SMS.

8.1.1. SMS Send Program

When an SMS send is selected and the launch button is pressed, the window as shown in Figure 2 will appear asking us to enter the phone number.

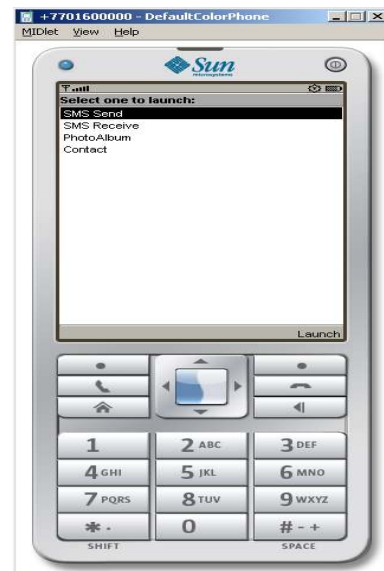


Figure 1. System implementation.

We can exit this window returning to the previous display as shown in Figure 1 by pressing the exit button. If no number is entered and the ok button is pressed, a new window will appear to indicate an error invalid phone number. While if a correct number is entered and the ok button is pressed, a new window as shown in Figure 3 will appear requesting to input the

message. If we press back in this case, the program will return to the previous display as shown in Figure 2. If the send button is pressed, the window as shown in Figure 4 will appear. Pressing no will cause the program go back from sending to Figure 1. Pressing yes will send the message to the destination as shown in Figure 5.



Figure 2. Input no.



Figure 3. Input message.



Figure 4. Request conn.



Figure 5. Sending.

8.1.2. SMS Receive Program

The receiving port must be opened by the receiver before a message can be sent since this proposed operating system is not a standalone operating system as shown in Figure 6. If exit button is pressed, the display will return to Figure 1.



Figure 6. Receiving.

8.2. Photo Album Manager Program

The photo album program is responsible of displaying photos. If photo album is selected and the launch button is pressed, the window as shown in Figure 7 is appeared.



Figure 7. Photo album choices.

8.2.1. JPEG Color

Display colored photos with Jpg extension Figure 8. Pressing back cause returning to the previous display as in Figure 7. if the option button is pressed, the program will display the options (Speed, Style) as shown in Figure 9 and any of the options can be saved by pressing ok and afterwards the program returns to previous display. The command option works with all of the choices.



Figure 8. Colored photo with jpg.



Figure 9. Speed control.

8.2.2. JPEGBW

Display black and white photos with Jpg extension as shown in Figure 10.

8.2.3. Jpeg Progressive

Display enhanced photos with Jpg extension.

8.2.4. Flashing Light House

Display moving images and can be controlled by the speed option, as shown in Figure 11.

8.2.5. Duke

Display message (Alert) indicating that the extension doesn't exists in the source.

8.2.6. PNG image

Requests the internet connection as shown in Figure 12. Pressing yes will connect to internet while pressing no will return to Figure 7.

8.2.7. Test chart

Displays the following chart as shown in Figure 13 .



Figure 10. Jpegbw.



Figure 11. Flashing.

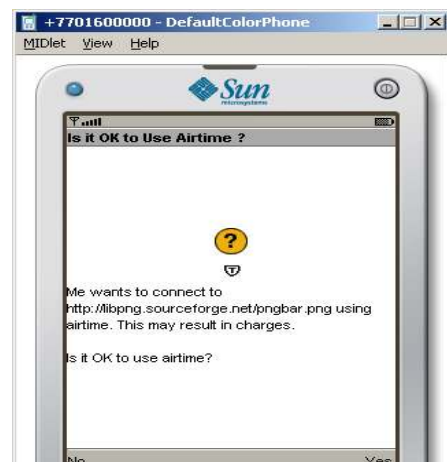


Figure 12. Internet connection.

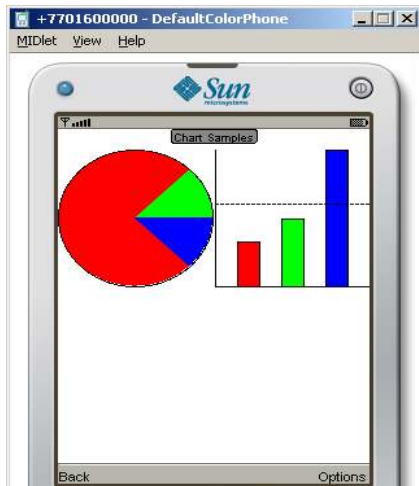


Figure 13. Test chart.

8.3. Contact Manager Program

The contact program displays person names storing in mobile phone, as shown in Figure 14, pressing ok to the selected name will display the phone numbers, email, and address, as shown in Figure 15.



Figure 14. Person names.



Figure 15. Name's detail.

9. Conclusions and Future Works

A mini-operating system has been built for managing some features and applications in mobile phone. A hard work is done in order to implement multitasking-multithreading execution based on this operating system. There was a need to build a basic class to connect all the applications designed to work on under the environment of this operating system. This proposed operating system was built and tested using java wireless toolkit and then embedding to mobile phone. This work helps to enter modern programming arena represented by programming miniature devices. Also, this work helps to keep in pace with the development in the field of programming embedded programs. In addition acquire the perquisite knowledge for these kinds of operating and management systems especially when developing special applications.

It is concluded that java 2 micro edition language support the multithreading in better work then other languages. So, using the multitasking with multithreading in this design does not need extensive memory management. Since the threads within a process share the same data and code address space, and most other resources. It is concluded that executing this proposed operating system by a virtual machine Java Virtual Machine (JVM) isolating it from native access and they can access only Java libraries or the functions of virtual machine. Also, dealing with mobile phone required a consideration to modern techniques in order to produce a responsive operating system for this kind of devices.

It is more an important to give suggestions to build completed applications for cellular mobile phone in addition. Build applications which connectable with hardware for cellular mobile phone in order to make full operating system with it's ability of executing and tabling.

References

- [1] Aciicmez O., Zhang X., and Seifertetal J., "Building Efficient Integrity Measurement and Attestation for Mobile Phone Platforms," in *Proceedings of the First International ICST Conference on Security and Privacy in Mobile Information and Communication Systems (MobiSec)*, Berlin, pp. 71-82, 2009.
- [2] Chitu A., "Google Launches Android, an Open Mobile Platform," available at: <http://googlesystem.blogspot.com/2007/11/google-launches-android-open-mobile.html>, last visited 2009.
- [3] Digia I., *Programming for the Series 60 Platform and Symbian OS*, John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, 2003.

- [4] Feng X., Shao Z., Dong Y., and Guoetal Y., "Certifying Low-Level Programs with Hardware Interrupts and Preemptive Threads," *Technical Report*, Department of Computer Science, Yale University, New Haven, CT, 2007.
- [5] Gogging G., "Adapting the Mobile Phone: The iPhone and its Consumption," *Journalism and Media Research Centre*, vol. 23, no. 2, pp. 231-244, 2009.
- [6] Gold J., Choosing an Enterprise-Class Wireless Operating System: A Comparison of BlackBerry, iPhone Associates, LLC., available at: <http://www.jgoldassociates.com>, last visited 2009.
- [7] Jiannong C., Feng X., Lu J., Chan H., and Das S., "Reliable Message Delivery for Mobile Agents: Push or Pull? IEEE Transactions on Systems", *Man, and Cybernetics, Part A*, vol. 34, no. 5, pp. 577-587, 2004.
- [8] John S., Dean M., and George Z., Power-Sensitive Multithreaded Architecture, Intel Corporation RA2-4012501 N.W. 229th Ave Hillsboro, 2000.
- [9] Jonatan W., "Comparison of CPU Management in Symbian OSTM and Microsoft R WindowsR CE," available at: http://www.ida.liu.se/~TDDB72/rtproj/reports2006/20-jonwa-mikos290-Symbian_WinCE_CPUmanagement.pdf, last visited 2006.
- [10] Maarten S., "Developing Multithreaded Applications for the NET Compact Framework," *MSDN Technical Article*, 2005.
- [11] Manisha M., "Multithreading Part I: Multithreading and Multitasking," *Technical Document*, National Instruments, 2002.
- [12] Morris M., "Mobile Operating Systems and the Military: Why Windows Embedded CE?," *Technical Document*, Director of Engineering Services BSQUARE Corporation, 2008.
- [13] Paul A., "Mobile and PDA Technologies: Looking Around the Corner," *Technical Document*, JISC Technology and Standards Watch, 2005.
- [14] Prasad K., *Embedded Real-Time Systems: Concepts, Design and Programming*, Black Book, DreamTech press, New Delhi, 2005.
- [15] Roceanu I., "Knowledge Anywhere, Anytime Based on the Wireless Devices," in *Proceedings of the 5th Scientific Conference eLearning and Software for Education*, Bucharest, pp. 68-72, 2009.
- [16] Schiffman J., Zhang X., Gibbs S., Kunjithapa A., and Jeong S., "Securing Elastic Applications on Mobile Devices for Cloud Computing," in *ACM Cloud Computing Security Workshop*, USA, pp. 127-134, 2009.
- [17] Shah B., James C., Hui D., and Jing D., "MANTIS OS: An Embedded Multithreaded Operating System for Wireless Micro Sensor Platforms," *ACMKluwer Mobile Networks & Applications (MONET) Journal*, Special Issue on Wireless Sensor Networks, vol. 10, no. 4, pp. 563-579, 2005.
- [18] Silberschatz A., Galvin P., and Gagne G., "Operating System Concepts (7th edition)," *Mc-Grawhill*, 2005.



Dhuha Albazaz is an assistant professor at Computers Sciences Department, College of Computers and Mathematics, University of Mosul. She received her PhD degree in computer sciences in 2004 in the speciality of computer architecture and operating system. She supervised many Master degree students in operating system, computer architecture, dataflow machines, mobile computing, real time, and distributed databases. She has three PhD students in FPGA field, distributed real time systems, and Linux clustering. She also leads and teaches modules at both BSc, MSc, and PhD levels in computer science. Also, she teaches many subjects for PhD and master students.