Volume 1, No. 2, July-August 2010

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

Luminosity Mass Entrenched Operating System Design for Cellular Phone

Pughazendi.N*
Faculty, Department of Master of Computer Application
Panimalar Engineering College
Chennai, India
pughazendi@yahoo.com

Balasundara Ganapathy.N
Faculty, Department of Master of Computer Application
Panimalar Engineering College
Chennai, India
balabsg@gmail.com

Abstract: Light Weight Embedded Operating System is used for tagging up the objects for mobile agent and to analyse the need of memory and process constraints in mobile phones. To analyze, study and design the several design issues such as Architectural Framework, Memory Management, Process Management, Network support, and the impact of Hardware on the operating system. The Application system scheduling, process management, security system designs are taken into consideration to implement the operating system design. The project uses Embedded Linux platform to implement the system design and to implement the system design in J2ME as user interface platform. The main analytical design over this system is to consumption of memory management and to design the patterns for further improving the performance and reducing the cost and reliability.These application case studies consists of and to design the light weight operating system to meet the mobile phone which are in mobile terminal consists of J2ME interfacing unit**.**

Keywords: Embedded, Mobile, Memory Management, Operating System, Protection.

I. INTRODUCTION

The need to support the safe execution of untrusted programs in runtime systems for type-safe languages is clear. Language runtimes are being used to execute untrusted code that may violate a system's safety or security. Current runtime systems implement memory protection in software through the enforcement of type safety. They do not, however, sufficiently isolate untrusted code and are unable to control the computational resources used by such code. We describe VirtualOS, a design for a Java virtual machine (JVM) that allows for the robust execution of untrusted code. Like hardware- based operating system that supports and manages multiple processes running on one physical machine, VirtualOS provides resource control and isolation to its processes.

Consider the following application scenarios that employ type-safe languages:

Java applets enjoy widespread use. There is no prior trust relationship between the originators of an applet—who could be a malicious attacker—and the client who executes the applet. A Java program can be verified to en- sure that the code will not compromise security on the client's machine, but it includes no defenses against denial-of-service attacks directed against computing resources such as memory and CPU. Even though Java was first released to the public in 1995, industrial browsers still do not withstand even the simplest of attacks.

Java has also become popular for many server-side applications, such as Java Server Pages or Oracle's JServer environment. Even though such code is usually trusted, a buggy application could cause the server to spend all its time collecting garbage and deny service to other applications.

Extensible operating systems allow applications to download code into the kernel. In the SPIN extensible OS, both the kernel and the extensions are written in the type-safe language Modula-3, which enables extensions' access to kernel interfaces to be controlled. However, it is impossible to control the resources used by a given extension—for instance, to guarantee that one extension obtains a certain share of CPU time.

These applications require a runtime system that supports the following features:

A. Protection

Protection includes confidentiality and integrity. Confidentiality requires that an application must not be able to read another application's data unless permitted. Integrity requires that an application must not be able to manipulate the data of another application or system-level data in an uncontrolled manner, or destroy the data of another application.

B. Isolation.

Applications must be isolated from each other. One application's failure must not adversely affect unrelated applications or the system itself.

C. Resource management.

First, resources allocated to an application must be separable from those allocated to other applications to ensure proper accounting. Second, resource management policies must guarantee that a un- privileged or untrusted application is not able to starve other applications by denying them resources.

D. Communication

Since the system may include multiple cooperating applications, applications should be able to communicate with each

other. For Java, an efficient way of sharing data should be supported that does not compromise protection and isolation.

These features are provided by traditional operating systems through processes. They support the level of assurance that a system administrator would want to execute completely untrusted code. In certain situations, greater levels of trust may be permissible, in which case some of these features could be relaxed. However, in situations where untrusted code is to be executed, we believe that any relaxation would be undesirable.

Existing mechanisms such as type safety, language-based access control, and permission checks provide protection—our research shows how to support the remaining three features. Some existing systems superimpose an operating system model on Java, but do so without changing the underlying virtual machine. Separating different applications is to give each one its own virtual machine and run each virtual machine in a different process on an underlying OS. Depending on the operating system have multiple drawbacks:

Systems with this structure cannot account for resources that are spent by the JVM itself. An alternative approach to the per-JVM overhead is typically high, and the flexibility with which resources can be managed may be limited. For instance, a typical JVM's memory footprint is on the order of 1–2 MB, which can severely restrict scalability. A JVM's startup costs, which include the cost of loading and linking the Java byte code, are typically high. When different instances of a JVM run on the same machine, they typically do not share any runtime data structures, even on systems that provide support for shared memory. Finally, the option of dedicating one JVM process to each application does not work on small devices that may not provide OS or hardware support for managing processes , or in software environments where the JVM is nested inside another application.

II. PROBLEM DEFINITION

The Application system scheduling, process management, security system designs are taken into consideration to implement the operating system design. The project uses Embedded Linux platform to implement the system design and to implement the system design in J2ME as user interface platform.

We focus on the three areas in the evaluation of our Virtual OS prototype. First, we measure the run time overhead Virtual OS introduces, when compared with a JVM that does not provide robust support for multiple processes. We show that the overhead is reasonable (from 0% to25%) when compared to the Kaffe VM upon which Virtual Os is based. Second, we evaluate Virtual OS's effectiveness in handling untrusted or buggy applications that may engage in denial-of-service attacks. Virtual Oscan defends against denial-of-service attacks directed at memory, CPU time, and the garbage collector; Virtual OS's integrity is not compromised by such attacks. Third, we compare Virtual OS to the possible alternative of running multiple applications on multiple JVMs, and show that Virtual OS can out scale that approach.

Our prototype runs as a user-mode application on top of the Linux operating system. All our measurements were taken on a 800 MHz "Katmai" Pentium III, with 256 Mbytes of SDRAM and a 133 MHz PCI bus, running Red Hat Linux.

III. SYSTEM OVERVIEW

We have presented the design and an implementation of a Java runtime system that supports multiple, untrusted applications in a robust and efficient manner. We have used operating system ideas to introduce a process concept into the Java virtual machine. Virtual OS's processes provide the same properties as operating system processes: they protect and isolate applications from each other, and they manage and control the resources that they use.

A. Existing System

Existing system consists of a Software interfacing units where access the Memory and Process constraints and their corresponding active objects which are high usage in Process Oriented Architecture [POA]. POA calls active objects, so response time is unpredictable, therefore service time is not considered. Present system loads or installs the software in the runtime, so it takes long time to respond. High current flow while loading the active objects.

Disadvantages of Existing System

- Existing system calls active objects which are in high usage in process oriented architecture.
- Existing system uses software interfacing units.
- Symbian OS is not suitable for phones with low RAM and slow processor.
- High current flow while loading the active objects.
- When calling the active objects response time in unpredictable and therefore service time is not considered.

B. Proposed System

Analysis the case studies and to enable the Process and Memory constraints for Runtime environment to enable the system architecture to increase the Memory and Process constraints. It enables the process constraints and memory during runtime environment. Proposed system is used to route to the object memory and process. Efficient use of RAM, memory addressing. If we reduce memory weight automatically it provides efficiency and perform efficient memory mapping.

Advantages of Proposed System

- Symbian OS using SIS format, it is third party software in which anybody can develop software in this OS.
- Symbian OS offers rich User Interface.
- Proposed system reduces memory weight and provides efficiency automatically and perform efficient memory mapping.
- They can be personalized the way the user wants by installing proper software.
- They support many types of security-applications like anti-theft and anti-virus applications.
- They support applications as advanced as PC applications, far advanced the Java (but they support Java too).

IV. GUI INTERFACE (MAIN APPLICATION)

A. Compass

A compass is a navigational instrument for determining direction relative to the earth's magnetic poles. It consists of magnetic pointer free to align itself with earth's magnetic field. A compass can be used to calculate heading, used with a sextant to calculate latitude, and with a marine chronometer to calculate longitude. It has been supplanted by modern devices such as Global positioning System [GPS].

B. Position

Longitude: It is the geographic coordinate used for global navigation for east-west measurement. It is an angular measurement ranging from 0 degree at the prime meridian to +180 degree eastward and -180 degree westward.

Latitude: Latitude gives the location of a place on earth north or south of the equator. Latitude ranging from 0 degree at the equator to +90 degree north pole and -90 degree south pole.

UTM: It shows the current location of the series 60 GPS phone in several grids coordinates. It is used to import map images to see your location.

C. Navigation

Navigation is the process of reading, and controlling the movement of a craft or vehicle from one place to another. This is one of the main principles behind the GPS system. The GPS system can tell you your location anywhere on or above Earth to within about 20 to 30 feet. Even greater accuracy, usually within less than three feet, can be obtained with "differential corrections" calculated by a special GPS receiver at a known fixed location.

D. Trip Computer

Trip Computer is personal off-road GPS navigation utility software for Google Android phone that has been created for active people, who love travel by hike or bike. Application does not use Internet connection or Google Maps for navigation or track recording, but you can share your data to others, or store your data on server. All you have is your own map view with bearing, distance to targets, your recorded tracks, orientation waypoints and compass with directions.

E. Device

The device icon contains two options device and satellites. This is used to find the mobile is connected with any personal computer or it is connected with the general packet radio service.

F. Settings

The settings option is used to track the precious location visited, used to get map, alert messages. This is also used monitor the speed and to send location and the connectivity of the mobile device

Figure 1 : Showing list of icons in main module

G. Search by Name

Search by Name icon is used to search for the specific type of business available in that location. Enter the first three letters of the business name and specify the location to search. If available it shows the list of addresses, if not it display the error message.

H. Search by Type

Search by Type icon uses specific category to search. For example we can search for automotive, entertainment, medical, restaurant, services, stores, travel. Otherwise we can also create a category to search.

I. Search for People

Search for people option is used to identify the person's address. For that need to specify the first three letters of the person's first name and last name and the location. It shows the list of persons available in that location. Then we can make call to that person, or we can get map and directions for that person's address.

Figure 2: Super pages showing list of icons

J. Weather

Weather icon is used to get weather report for the specified location. Specify the location and then choose search option. It shows the weather report for the specified location for next five days.

K. Maps

Maps icon is used to get route map for the specified location. Choose location and then choose search, it will show the route for the specified location.

L. Driving Directions

Driving directions icon is used to get directions for address we specified. Specify from-to location and choose search option. This shows the available for that specified direction.

V. PROGRAM DESIGN LANGUAGE

A. Search by Name

```
INPUT  Business name and Location from the user
IF  Business  name is valid THEN
        SET  location to search
        DISPLAY  business name  AND corresponding
address
ELSE  Business name is not valid
        DISPLAY No listing found with that name, try with
some other name
        END IF
```

B. Search by Type

```
INPUT Category name to search from the user
IF Category is valid THEN
        DISPLAY Types available in that catogory
ELSE IF
        SET  new category to search
        DISPLAY Types available in that category
ELSE IF  specified category is not valid THEN
        DISPLAY No field available in that category. Search
someother category
END IF
```

C. Search for People

```
INPUT First letter of the Firstname and Lastname of a person
to search
IF Firstname and Lastname is valid THEN
        SET  location to search
        DISPLAY person name AND corresponding address
THEN
        DISPLAY map,directions and calling option for that
persoon
ELSE IF data is not valid THEN
        DISPLAY Enter atleast three letter of  the firstname
and lastname of a person
        ELSE IF data is not valid THEN
                DISPLAY No person available with that
name
        END IF
```

D. Search by Phone Number

```
INPUT Phone number of the person to search
IF  phone number is valid THEN
        DISPLAY person name AND calling option
ELSE IF phone number is not valid THEN
        DISPLAY Enter full ten digit of the number
END IF
```

E. Weather

```
INPUT Location from user to search
IF  location is valid THEN
        DISPLAY Next five days Weather  report for the
specified location
ELSE location is invalid THEN
DISPLAY Location not found, try some other location
END IF
```

F. Maps

```
INPUT Location from user to search
IF  location is valid THEN
        DISPLY route map AND available direction for the
specified location
        DISPLAY zoom-in,zoom-out,left,right option for that
location
ELSE IF location is invalid THEN
        DISPLAY No listing found, more specify the
location
END IF
```

G. Driving directions

```
INPUT From-to location from the user to search
IF location is valid THEN
        DISPLAY directions for the specified location
ELSE IF  loation invalid THEN
        DISPLAY No directions found. Search from some
other location
        SET direction for the location
ELSE
        DISPLAY directions for the specified location
END IF
```

V.CONCLUSION AND FUTURE ENHANCEMENTS

We have demonstrated that separating “kernel” code from “user” code is necessary to protect critical parts of the system from corruption when applications are killed. Code that executes in kernel mode delays termination requests until it returns to user mode; in addition, it is written such that it can safely back out of exceptional situations. This separation does not prevent sharing, because it does not change the way in which objects access each other’s fields and methods; type safety remains the means to enforce memory safety.

We have presented the design and an implementation of a Java runtime system that supports multiple, un trusted applications in a robust and efficient manner. We have used operating system ideas to introduce a process concept into the Java virtual machine. Virtual OS's processes provide the same properties as operating system processes: they protect and isolate applications from each other, and they manage and control the resources that they use.

VI. REFERENCES

[1] Tullmann, P., Hibler, M., and Lepreau, J. 2001. Janos: A Java-oriented OS for active network nodes. IEEE J. Sel. Areas Communication. 19, 3 (Mar.), 501–510.

[2] Tullmann, P. and Lepreau, J. 1998. Nested Java processes: OS structure for mobile code. In Proceedings of the 8th ACM SIGOPS European Workshop (Sintra, Portugal). ACM, New York111–117.

[3] Van Doorn, L. 2000. A secure Java virtual machine. In Proceedings of the 9th USENIX Security Symposium (Denver, Col.).USENIX Association,19–34.

[4] Engler, D. R., Kaashoek, M. F., and O'Toole J R., J. 1995. Exokernel: An operating system architecture for application-level resource management. In Proceedings of the 15th Symposium on Operating Systems Principles (Copper Mountain, Col.). 251–266.

[5] Flatt, M. and Findler, R. 2004. Kill-safe synchronization abstractions. In Proceedings of the SIGPLAN '04 Conference on Programming Language Design and Implementation (Washington, D.C.). ACM, New York.

[6] Ford, B., Hibler, M., Lepreau, J., Tullmann, P., Back, G., Andclawson,S. 1996. Microkernels meet recursive virtual machines. In Proceedings of the 2nd Symposium on Operating Systems Design and Implementation (Seattle, Wash.). USENIX Association, 137–151. FRANZ, M. 1997. Beyond Java: An infrastructure for high performance mobile code on the World Wide Web. In Proceedings of WebNet '97, World Conference of the WWW, Internet, and Intranet,S. Lobodzinski and I. Tomek, Eds. Association for the Advancement of Computing inEducation, Toronto, Ont.,Canada, 33–38.

[7] Gay, D. and Aiken, A. 1998. Memory management with explicit regions. In Proceedings of the SIGPLAN '98 Conference on Programming Language Design and Implementation (Montreal, Ont., Canada). ACM, New York, 313–323.

[8] Dillenberger, D., Bordawekar, R., Clark, C. W., Durand, D., Emmes, D., Gohda, O., Howard, S., Oliver, M. F., Samuel, F., and John, R. W. S. 2000. Building a Java virtual machine for server applications: The JVM on OS/390. IBM Syst. J. 39, 1, 194–210.

[9] Domani, T., Goldshtein, G., Kolodner, E. K., Lewis, E., Petrank, E., and Sheinwald, D. 2002. Thread-local heaps for Java. In Proceedings of the 3rd International Symposium on Memory Management (Berlin, Germany). ACM, New York, 183–194.

[10] Dorward, S., Pike, R., Presotto, D. L., Ritchie, D. M., Trickey, H., and Winterbottom, P. 1997. The Inferno operating system. Bell Labs Tech. J. 2, 1, 5–18.