Design and Implementation Considerations for a Pedagogical Object Oriented Operating System

MTech Thesis, Jan 2009

Pinaki Chakraborty
School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi – 110067, India
E-mail: pinaki_chakraborty_163@yahoo.com

For centuries, experimentation has been at the center of all successful scientific studies. Experimental explorations can offer new perceptions, weed out unproductive approaches and validate theories and practices. Computer Science (CS), as a discipline, has a poor record when it comes to experimental validation. The ratio of theoretical research claims that have not been experimentally verified is notably high in CS. The situation is no different in the field of system software where experimental research has now become a necessity. The thesis in discussion presents an experimental study on operating systems. The thesis first develops a design of a new operating system, called JNUOS, imparting several noble ideas and then provides a commentary on its implementation [1-7].

The operating system has been developed following a few well established paradigms and models of CS [2, 6]. The operating system has been developed to serve as a pedagogical tool for the courses on operating systems [7]. It has been designed to have a microkernel based architecture. Among all microkernel based architectures, the multiple server microkernel based architecture has been chosen because of its superior modularity [1]. Object oriented methodologies have been then used to obtain a proficient implementation of the operating system [5]. Moreover, the development process has been guided by principles from software engineering and software architecture.

The operating system has a modular and stratified design with five distinct layers comprising of system components and application programs (Figure 1). The lowest layer consists of the microkernel and the clock driver. The second layer contains the drivers for common input/output devices including keyboard, display, mouse and printers. In contrast to most operating systems, these device drivers execute in an unprivileged user mode and can access the computer hardware only through the microkernel. The third layer contains system components for consolidation and optimization of services provided by the layers below it. These components include a teletype driver and a message type reader component. The fourth layer contains several servers each providing a particular type of service. These servers include the process manager, the information server, the file server, the login server, the reincarnation server and the verbose server. The fifth layer contains the shell, inbuilt utility programs and application programs. The programs in the fifth layer are free to make any number of system calls to any of the servers in the fourth layer. The servers, on their part, service these request and thus complete the client-server architecture.

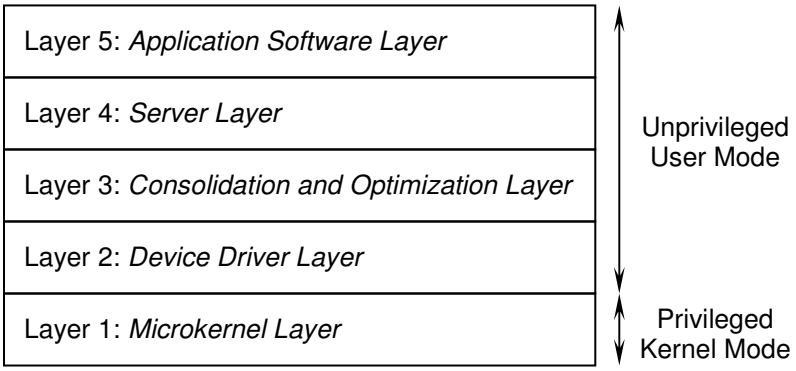

Figure 1: The multilayered architecture of the JNUOS operating system.

The concept of the reincarnation server has been recently introduced by some researchers to enhance the robustness of operating systems. This thesis attempts to augment the concept. The reincarnation server used in this operating system has the authority to correct all malfunctioning system components in the second, third and the fourth layers including itself. The reincarnation server detects malfunctioning components and replaces it with a fresh copy from the disc. The reincarnation server prompts the user before replacing a system component so that it may not replace any system component that, according to the user, is working satisfactorily.

The thesis introduces the concept of verbose mode of operation of an operating system [3]. The verbose mode facilitates the users to learn more about the internal working of the operating system. The verbose mode has been implemented primarily using three components, viz., the information server, the verbose server and the explanation module. While the first two components have been placed in the fourth layer, the explanation module has been placed in the fifth layer of the operating system. The verbose server and the explanation module have been developed only to realize the verbose mode. However, the information server is used for various purposes and is invoked by different components including the reincarnation server. The verbose server maintains a log of all the important events occurring in the computer system and also collaborates with the information server to create a portrayal of the operation of the entire operating system. This portrayal, or a part of it, is available to any user program on demand. The verbose mode is then completed by the explanation module that probes the verbose server and explains its replies to the users in a stepwise manner. The verbose mode has been tested for providing information of various types and varied depths. The verbose mode typically reports all process related activities including creation, termination and scheduling. Moreover, the output of the explanation module can be customized.

The operating system has been implemented predominantly in the C++ programming language. The C++ programming language has been chosen because it is an object oriented programming language that is well known for its ability to implement system softwares. To realize the different constructs of the operating system, twenty-three different classes have been first defined [5]. All important entities in the operating system have been then modeled as objects of these classes. Well accepted object oriented programming features have been used to implement the operating system. The concepts of inheritance and delegation have been used to define the related classes. Both static polymorphism, in the form of function overloading, and dynamic polymorphism, in the form of runtime function dispatch, have been used to implement the operating system. Constructors and destructors have been defined for most classes. Some of these classes even have overloaded constructors. The size of the source code of the operating system is approximately 6 KLOC.

The operating system supports a character user interface and a simple graphical user interface. The shell of the operating system is a character user interface based command line interpreter program that recognizes twenty-nine commands of varied types [4]. These commands can be broadly classified into six categories according to their purposes. These categories are those of the file related commands, directory related commands, user related commands, clock related commands, information related commands and miscellaneous commands. Certain commands can be issued using multiple syntaxes and have aliases. Each of these syntaxes and aliases can be considered as a variant of that command. There are a total of fifty variants of the twenty-nine commands. A preliminary testing of the behavior of the operating system has been carried out by issuing different permutations of the commands under various circumstances and results have been found to be satisfactory.

The operating system has demonstrated the feasibility of amalgamating the concepts of microkernel and object oriented programming. The overall behavior of the operating system has been also found to be satisfactory. Thus, the JNUOS operating system verifies the concepts and the design professed in the thesis.

ACKNOWLEDGEMENTS

Late Prof. R. G. Gupta and Prof. P. C. Saxena supervised the research presented in this thesis. The research was partly supported by a research fellowship from University Grants Commission, New Delhi.

REFERENCES

[1] P. Chakraborty and R. G. Gupta, “A structural classification and related design issues of operating systems,” *Proceedings of Second National Conference on Methods and Models in Computing*, 2007, pp. 265-273.

[2] P. Chakraborty and R. G. Gupta, “The design of a pedagogical operating system,” *Proceedings of Second National Conference on Computing for Nation Development*, 2008, pp. 517-527.

[3] P. Chakraborty, “Verbose mode of operation of a pedagogical virtual machine operating system,” *Proceedings of Third National Conference on Methods and Models in Computing*, 2008, pp. 43-53.

[4] P. Chakraborty and P. C. Saxena, “A comparison of the shell commands of three contemporary operating systems,” *Proceedings of National Conference on Modern Trends in Information Technology*, 2009, pp. 89-95.

[5] P. Chakraborty and P. C. Saxena, “The object model of the JNUOS operating system,” *Proceedings of Third National Conference on Computing for Nation Development*, 2009, pp. 89-95.

[6] P. Chakraborty, “A model of the computer operating systems,” *Journal of Management and Information Technology*, 2009, **1**(1): 83-95.

[7] P. Chakraborty and P. C. Saxena, “Novel approaches to teach and learn courses on computer operating systems,” *Computer Science and Telecommunications*, 2009, **8**(5): 174-176.