On Making the Theory of UNIX more Practical

Justus S and S. K. Ramalingam
K.L.N. College of Information Technology,
Pottapalayam, TN, India

justus.mku@gmail.com

Abstract

Information technologies are experiencing an exponential growth rate, whereas the academia tends to display a disproportionate approach in teaching the subject. In many Engineering institutes, it is observed that in the teaching-learning process of the theory, sufficient stress is not given to the related practical aspects. The result is that the students fail to get the 'feel' or 'touch' of the subject. This setback in learning often produces graduate products that the industry is reluctant to recruit.

This piece of work attempts to introduce a more practical learning approach for teaching UNIX. A new restructured series of lab exercises is introduced for the theory of Unix and network programming. An evaluation scheme is also designed to assess the proposed practical assignment and its effectiveness on the exposure gained by the students in moving towards open source tools and operating systems.

Keywords: Unix, network programming, curriculum, Practical exercises.

Introduction

The modern style of learning in computer science and information technology has seen a remarkable change. Though the change is constructive, its outcome is too averse to be on a par with the information technology industry (S & Alagarsamy, 2006; Kautz & Thaysen, 2000). The technology providers are inventing newer tools and technologies so rapidly that the academia feels a kind of disproportion in their advancement along with the industry. The setback in the learning process of the institutes more or often produces graduate-products, which the industry is reluctant to recruit. So it is necessary on the part of the institute to train up the students and impart in them a practice-oriented approach for learning. As software development has already moved towards open source tools and platforms, the process of learning too has to be practiced in an open-minded and experiment-based methodology (Gandhi, Shetty, & Shah, 1992; Ras, Avram, Weibelzahl, & Waterson, 2005). Correspondingly, the practical sessions of the theory have to be more industry-based, so that the learning community will be benefited and will get trained for its career and future prospects.

A study on the subject listing offered by the universities in India showed that many job-oriented courses are on the anvil. Amongst which, the authors found UNIX as a subject of learning where theory could practically be implemented in laboratories and in real-time platforms. From UNIX shell programming to UNIX socket and network programming, the subject of study is a pro-

gramming based theory which demands stringent programming practices.

The following section outlines the background of this research work by providing a detailed picture on the curriculum offered and its connection to research. In the third section, a detailed discussion on making the theory of Unix more practical is presented. A set of newly restructured, industry based lab practice exercises is proposed. The fourth section introduces a new evaluation scheme designed to assess the performance of the students and validates the authenticity of the lab exercises. Finally, the future scope of this work is dealt in the fifth section.

Background

Moore's law is applicable even in the growth of IT enterprises. Hence, the pace of the learning processes requires tremendous increase and demands reengineering from time to time (British Columbia, 2005; Kautz & Thaysen, 2000; Ras, Memmel, & Weibelzahl, 2005). Universities and autonomous colleges are considering reengineering and restructuring of curriculum for post-graduate studies time and again. This sort of curriculum upgradation helps the master degree computer science students to face the IT industries' requirements and demands. Being in the community of teaching and having close interaction with the IT industry, we have understood that some of the subjects taught at the postgraduate level need more practical-oriented coaching. As the IT corporates speak on the execution of cross-platform software products, most of them are opting for UNIX or one or the other flavors of LINUX. Research houses also prefer UNIX based environment for learning. This is due to the well-appreciable features available in the family of UNIX (Stevens, 1999; Stevens, Fenner, & Rudoff, 2003a, 2003b).

'Unix and network programming' offered by Anna University, Chennai, India, for the postgraduate computer applications course is quite interesting to analyze in a research perspective and interpret its significance from the industrial perspective. Out of the 256 engineering colleges affiliated to the University, approximately 60-70 percent of the colleges offer postgraduate program in computer application. Apart from this University there are 11 other Universities offering Unix as core subject in their post graduation program. This is also extended to the affiliated colleges. Hence the area under discussion, UNIX and programming practice, has a wide scope of appreciation. The motivation for this research work is embedded in the need felt by the teachers who handled the subject.

The postgraduate program in computer applications is spread over six semesters, out of which the first five are in-campus study and the last semester offers off-campus project work. The subject code CA234 is the theory of UNIX ad network programming and CA 235 is the practical for theory, both offered in the third semester. Three sets of postgraduate students have studied under this curriculum. It was then reengineered during 2005 and imparted for 2005-2006 batch of students, in the fourth semester, and is prevailing till date, with the subject codes MC1751 and MC1756, for theory and practical, respectively. Some of the terminologies used in this paper are: Program-the degree program, Course-the discipline, Subject – a specific area of study (offered with subject code), UNIX – the operating system, Unix – the subject of study / the programming environment, Teachers – Lecturer/Senior Lecturer/Asst. Professor/Professor.

Practical Based Curriculum

With that sufficient discussion on the nature of the subject and the psychology of the students, we move forward to propose a refined and reengineered curriculum for the postgraduate course on UNIX and network programming. We also attempt to integrate our research with the educational activities in these theory and practical subjects by presenting experimental evaluation processes. The evaluation methodology would be more feasible for its simplicity in implementation and its ability to measure the significance of the streamlined curriculum.

MC1751 – Unix and Network Programming

This subject is taught in the fourth semester of the postgraduate computer application program. It requires basic knowledge in Unix operating system, shell programming and kernel level functions, all of which are covered in the undergraduate courses offered by other Universities. All the engineering colleges and technical institutes affiliated to Anna University, Chennai, follow the same curriculum prescribed for a course by the University. Table 1 pictures the course plan for the subject MC1751.

Table 1. Course plan for MC1751

Week	Topics
1	Introduction to UNIX, File sharing, files and directories
2	File system calls, Streams, Standard I/O, Groups
3	Process management, Process controls, logins, signals
4	Intro to message passing, pipes, FIFOS
5	Message queues, read-write file & record locking
6	Semaphores, Shared memory
7	Intro to Transport layer, sockets, TCP, UDP, Raw
8	Applications of TCP, UDP sockets.

Forty five to sixty hours are planned for lectures leaving two to three weeks for internal tests and Units review. The prescribed reference books for the study of the subjects are UNIX (Stevens, 1999; Stevens et al., 2003a, 2003b). A detailed coverage on Unix filesystem, streams, buffers, process management, and related system calls, provide a better understanding of the advanced features in Unix programming environment (Stevens, 1999). An application oriented exposure on IPCs, starting from pipes, FIFOS, Message queues to shared memory, with SVR4 and postfix implementation, offer a good delivery on the theory part of process communication. The applications of IPCs are dealt under process synchronization using mutexes and semaphores (Stevens et al., 2003b). The most important aspect of network programming is through sockets. The power of Unix sockets, UDP and TCP sockets, raw sockets and their applications in network programming, are the topics covered under Units IV and V (Stevens et al., 2003a). The theoretical design issues for the development of applications in Unix environment provides the students with an extreme exposure to the programming techniques using C/Unix.

During the course of study, a variety of state-of-the-art concepts and techniques in UNIX are introduced to the students, either during the lecture hours or during lab hours, for added knowledge on the subject. In fact, the curriculum for MC1751 is highly technical and rich in its content presentation, except the teachers will have to have an organized strategy for time management, or otherwise schedule management, for the complete coverage of the subject without loosing the essence of the curriculum. Hence, teachers handling the subject will have to put in a lot of effort to prepare presentations, conceptually and practically, before a lecture session, and make the talk technically meaningful and convincing

MC1756 – Unix and Network Programming Lab

Experiencing the acquired knowledge is the complete form of learning (S & Alagarsamy, 2006). The attained knowledge is put into practice on the desks of laboratories. The lab, Unix and network programming lab (MC1756), specific for Unix and network programming (MC1751) is offered in the fourth semester under the revised regulations 2005. The revised lab exercises for the lab MC1756 turned up with a new approach, proposing one programming exercise for each topic,

covering advanced Unix programming, IPCs and socket programming. Certainly, the entire syllabus of MC1751 is covered. But the knowledge gained in the theory is questioned. One exercise per concept is not sufficient to gain practical experience on the studied theory because the student will not have a focus on the concept. The fundamental problem with the old exercises is that they lack in providing focused, hands on programming experience to the students, and no real time technical and operational problems are provided. An unstructured discussion among the students revealed that the old experiments did not give sufficient practical programming exposure to the theory learnt. The students also felt that they found it difficult to pass through the competitive exams due to the lack of programming knowledge imparted during the lab sessions.

The reengineered lab exercises that we intend to propose will cover the following: briefing on the basic Unix commands like system, dup, stat, signals, passwords, and shell commands for shell programming; all of them can be well comprehended by simple exercises, in two four-hours lab sessions. Then exercises using system calls for implementing file management and process management may be drafted. This session may consider four to eight exercises; covering create, open, read, write, close, stat, fstat, lseek for file management and fork/vfork, get/setpid, sessions, groups for process management. The notion of fork (child process creation) requires a worthy aptitude in managing processes through programming.

In continuation with process management, a rational understanding of process communication is essential. Pipes, FIFOS, message queues and shared memory are IPCs, which need problematic approach requiring moderate effort in solving them in an industrial perception. Programs in full-duplex multi process communication, FTP using message queues and distributed computing concepts in shared memory are suggested for practice. To have an exposure on semaphores, programs on process synchronization may also be provided. All these could be completed in a four four-hour lab sessions. The socket concepts may be started by providing basic message passing programs to chats and FTP programs, giving a practical knowledge on client-server concepts in network programming. The programs can further be extended by implementing concurrent servers using threads. Such socket programming knowledge and hands-on experience is preferred in telecom domain industries.

The comprehensive lab exercises for practical Unix is proposed in Table 2.

Table 2. Proposed lab exercises for MC1756

Week	Proposed Lab Exercises	No. of Exercises
1,2	Demo Exercises in basic Unix commands	4
3,4	Exercises using system calls for file management	3
5	Exercises using system calls for process management	5
6,7	Pipes and FIFOs (full-duplex, multiprocess communication)	3
8,9	Message queues (Queuing, FTP)	2
10	Shared memory (distributed computations)	2
11	Semaphores (process synchronization)	1
12	Message passing using TCP/UDP sockets (Echo server-client)	2
13-15	Chat Exercises using UDP sockets	1
16-17	Implementing FTP using TCP sockets	1
18	Concurrent servers using threads	1

The proposed lab exercises for MC1756 have considered a wider range of concepts in Unix programming environment. Sufficient numbers of exercises have been given for each topic covered in the theory. Now the student will have a variety of programming practice for each of the con-

cept; from file management to socket programming. The necessity for this kind of multifaceted lab exercises from the industry partners is discussed in the following sub-section, and the significance of the proposed lab exercises from the students' perspective is given in the fourth section, with satisfactory evaluation schemes.

Incorporation of IT industry and Institute

Experiment, according to the Oxford dictionary, is *a new course of action adopted to demonstrate a known fact (specification) without being sure of the outcome*. A real time problem specification, whose outcome is indeterminate in the early stages, adopts unique course of tasks to produce an end-product, is explained as Project (Duncan, 1996). From these definitions it is clear that experiments conducted in laboratories are subsets of projects carried out in IT companies. For this reason of correlation between the definitions, IT recruiters prefer fresh graduates and post graduates to have a practical exposure in programming skills under specific domains.

The pre-placement talk delivered by companies, like Wipro, HCL, IBM global services, Infosys, Carritor, Covansys and many more, coming for off-campus or on campus recruitment campaigns insists on the student community to have in-depth knowledge in the subjects as well as adequate hands-on programming experience, gained during the course of his/her study. This requirement will cause the recruiting company to spend less effort in offering training to the freshers, in their domain.

Comments: To legalize this proposed practical curriculum for Unix and network programming lab (MC1756), it is submitted to project leaders, software professionals and recruiting heads of a few companies in Chennai and Bangalore, for discussion. The excerpts from the comments are:

- The basic programs using Unix commands and system calls will offer a better understanding and provides a strong foundation for the future practical exercises. The newly proposed lab exercises covering concepts file management and process management will train the students in doing kernel level modular programming. Due weightage is also given to IPC programming which is sufficient for a fresher to gain an understanding of the concepts and their programming implementation. Network programs using TCP/UDP sockets offer satisfactory exposure and adequate prerequisite for router programming and other network related programming.
- The previous comment is accepted in full and additional comments are: 1) Algorithm implementation in IPC and semaphore programming may be included, 2) Internet based simple programs may be offered in the interest of students preferring web design, and 3) An application development may also be included at the end of Lab exercises, which will give a complete exposure in software development in Unix platform.
- Though the lab exercises of MC1756 are to the standards of the IT industry, the practicalities of the teachers are questioned for their expertise in the programming subject. Therefore, as a preliminary phase the teachers handling the subject MC1751 and MC1756 should be given an hands-on experience in the experiments proposed. The lab exercises offer rigorous programming practice and impart hands-on experience and knowledge in the subject.

The comments are a combination of both appreciation and suggestion from the experts in the subject and in academia. We regard these comments and proposition, and further, recommend the academic architectures and implementers to put them into operation in their forthcoming endeavors.

Evaluation Scheme

We track students' progress in the practical subject, MC1756, with detailed evaluation forms. The programs early in the practical sessions involve exposure to simple concepts, while those later will require students to work on the advanced features of Unix programming. This gradual increase in the complexity of learning is assessed in three phases covering multi-dimensional evaluation aspects.

Phase I

A. On the first day of the lab session, all students will be asked to fill out a two-part questionnaire. The first part will be designed to gather demographic information (race, gender, etc) as well as about the previous lab courses, especially in programming languages (C, C++, Assembly language and internet programming languages).

B. The second part of the questionnaire will ask students to indicate the current knowledge in the theory subject MC1751 and the understanding of the major topics like file management, process management, IPCs, process synchronization and sockets, to be covered in the lab sessions. Each participant will have an option to provide written comments.

Phase II

A. At the end of the semester or in the last laboratory session, the second part of the above-described questionnaire, with appropriate modifications to items on expectations and some new items on the lab exercises will be given to the students.

B. A summative technical test on the entire practical exercise will be conducted. The test consists of a combination of multiple choice, descriptive and performance-based type items to provide an accurate assessment of what the students know and are able to do.

Phase III

A. Information from the first part of the questionnaire will be used to know the background of the students enrolled for the lab. The information may also be used for document purpose and for statistical analysis in future.

B. Results from parts A and B of phase II will form basis for assessing the degree to which the students expectation of the practical exercises are met and how much they have gained knowledge in the subjects, both MC1751 and MC1756. Individual percentage score in the technical test and the overall score of the class are computed to evaluate the practical essence, including the tutorials and instruction classes/manuals.

C. Appropriate statistical analysis (Alagarsamy, 2007; Cohen, 1998) will be performed to identify the significant outcome of the students' learning process and the curriculum content. It should be noted that the sample size of a class of 60 may be small and hence univariate/multivariate statistical tests may not serve adequate power to detect significant differences (Tatsuoka, 1998). Hence, information collected from a sample of colleges, one from each zone, will facilitate the interpretation of the outcome.

The three-phase evaluation plan is designed based on the works done in assessing curriculum and students' ability to learn (Anatasi, 1988; Cohen, 1998; Tatsuoka, 1998).

Experiments

The second year post-graduate students of KLNCIT are studying the practical (MC1756) curriculum, given by Anna University. They are asked to fill-in the Evaluation Form-I (EF-I) at the start

of the course. In addition to the existing lab experiments we introduced our proposed lab exercises to 45 students. Only a few significant lab exercises, covering file management, process management, IPC and socket programming, are practiced apart from their regular practicals. At the end of the practical course each student was asked to fill in the Evaluation Form-II (EF-II). Careful and extra consideration was given by the students while filling the form, so that incorrect or irrelevant information are avoided during the evaluation process. The results are found encouraging.

Results

The plan is used to evaluate how the students, undertaking the courses MC1751 and MC1756, are specialized in the major areas covered both in theory and practical. Student's expectations and the essence of the lab exercises, in equivalent with the industry and theory, and how far they are realized and meted out are also evaluated. Result of EF-I shows that 40% of the students hadn't had previous programming experience and 87% of students haven't studied UNIX as subject in their under graduation. Most students preferred learning the practicals with sufficient demonstration. Assessing EF-II shows significant results on the students' understandability of the subject Unix. 55.56% of students' expectations are met in providing the new lab experiments and 44.45% of students deem that the proposed lab experiments are useful and are technically challenging.

Table 3. Results of the Evaluation

Topics	LU the old lab exercises	Standard Deviation	LU the new lab exercises	Standard Deviation	Improvement (%)	Standard Deviation
Basic Unix commands	4	0.10	7	3.12	32	30.17
File management	2	1.32	6	2.62	46	13.06
Process management	2	1.32	4	2.25	28	9.39
Pipes and FIFOs	5	0.81	8	3.07	36	22.68
Message queues	3	0.61	6	2.70	36	20.96
Shared memory	2	1.32	5	2.39	32	10.78
Semaphores	2	1.32	6	2.62	44	13.06
Message passing, sockets	6	1.51	9	3.10	39	15.91
UDP sockets	6	1.51	10	3.48	41	19.62
TCP sockets	3	0.61	4	2.37	15	17.63
FTP using TCP sockets	4	0.10	6	2.86	27	27.65
Concurrent servers	3	0.61	5	2.49	26	18.85
Web based programming	4	0.10	6	2.86	24	27.65
Application development	6	1.51	10	3.48	46	19.62
Overall understanding	6	1.51	9	3.10	34	15.91

LU – Level of Understanding

The results shown in Table 3 are the outcome Evaluation Form-II. The best responsive answers for each lab exercises are analyzed and critically assessed for their degree of understandability.

The level of understanding (measured against a scale of 10) of the current curriculum is compared with the proposed lab practical. The degree of improvement in understanding the subject is found to range from 15%-46%. This shows a considerable improvement in the significance of the proposed lab experiments in the context of coverage of the theory and the correlation with the industrial practice. Though the sample size is small, the questionnaires and the data collected in the three phases are consistent in all respects. This evaluation can be further extended to larger sample sizes by including other institutes offering Unix in their curriculum.

Conclusion

This piece of work is an effort to suggest the academic leaders and implementers of curriculum that the syllabi framed for postgraduate programs need regular restructuring and reengineering, catering to the demands of the IT industry. Though the theory part of Unix and network programming is quite well framed, its practical lab exercises were very preliminary and needed reconsideration. Hence we attempted to work out a new set of lab practical suite, or even a protocol, whatever it be named, to provide postgraduate computer application students with hands-on experience in the programming part of Unix. An evaluation scheme is also designed to assess the realism of the proposed curriculum for lab and whether the change in the lab exercises has really solved the purpose for which it was altered**.** We have also shown that our proposed practical exercises have the potential to extend the state-of-the-art technologies and provide a new pedagogical opportunity for the postgraduate application students as well.

The future issues with regard to this work involve conducting surveys in postgraduate and undergraduate programs, offering Unix as theory and practical subjects. The curriculum offered by other state Universities in India, and still to extend the scope, in Universities in the international arena may be analyzed. Other subjects like XML and web services (MC1801), Middleware technologies (MC1754) and Internet programming (MC1705) needed reengineering both in theory and in practice labs. Executive training for teachers offered by Anna University, companies interested in institute collaborations and privileged corporate trainers will be of direct benefit to the students. The tacit knowledge acquired from these training programs should be transformed to explicit knowledge through externalization (Alagarsamy, S, & Iyakutti, 2006, 2007). That is, the lab manuals, given during training sessions or prepared by the teachers should be made available online in the intranet/internet and in libraries (Ras, Memmel, & Weibelzahl, 2005). This system of learning has reaped significant results in learning organizations (Kautz & Thaysen, 2000; Vestal, 2005) and also promotes industrialization of institutes. This instance of effort to make Unix more practical will serve as a model for further research in regular reengineering and restructuring of the process and the system of study.

Acknowledgements

We acknowledge the support provided by the Management of K.L.N. College of Information Technology and the Head of the Institution in approving us to conduct experiments and surveys among their MCA students. Also the espousal given by the faculty of MCA is acknowledged. We also acknowledge the contributions of industrial partners, Sryas Software Pvt. Ltd, Chennai and Congent Technologies, Chennai, and Chris Software Solutions, Bangalore, India.

References

Alagarsamy, K., S, J., & Iyakutti, K. (2006). A theoretical perspective on knowledge based organizational learning. *13th APSEC*. Digital Library, IEEE Computer Society. Retrieved from www.ieeecomputersociety.org/10.1109/APSEC.2006.15

Alagarsamy, K., S, J., & Iyakutti, K. (2007). The knowledge based software process improvement program: A rational analysis. *2nd IEEE International Conference on Software Advances*, France.

Anatasi. (1988). *Psychological testing.* Macmillan.

Bloom, S., Hasings, J. T., & Madaus, D. F. (1971). *Handbook of formative and summative evaluation of student learning.* McGraw-Hill.

British Columbia. (2002). The future of employee learning: A learning strategy for the BC public service. Published in the Legislative Assembly of British Columbia. Retrieved from http://www.pserc.gov.bc.ca/els/pdf/Corporate%20Learning%20Strategy.pdf

Cohen, J. (1988). *Statistical power anaysis for the behavioral sciences.* Erlbaun Publishing.

Dodgson, M. (1993). Organizational learning: A re-view of some literatures. *Organization Studies*, *14*, 375-394.

Duncan, W. R., (1996). *A guide to the project management body of knowledge.* Newton square, PA, USA: PMI.

Gandhi, M., Shetty, T., & Shah, R. (1992). *The 'C' Odyssey Unix –The open boundless C* (1st ed.). BPB Publications.

Kautz, K., & Thaysen, K. (2000). Knowledge, learning and IT Support in a small software company. *Proceedings of BPRC CONFERENCE on 'Knowledge Management: Concepts and Controversies,'* pp. 10-11.

Kirk, R. E. (1995). *Experimental design: Procedures for the experimental sciences*. Brooks/Cole Publishing.

Quinn, R. G. (1993). Drexel's E^4 program: A different professional experience for engineering students and faculty. *Journal Engineering Education*.

Ras, D., Avram, G., Weibelzahl, S., & Waterson, P. (2005). Using weblogs for knowledge sharing and learning in information spaces. *Journal of Universal Computer Science*, *11*(3), 394-409.

Ras, E., Memmel, M., & Weibelzahl, S. (2005). Integration of e-learning and knowledge management – Barriers, solutions and future issues. *3rd Conference Professional Knowledge Management - Experiences and Visions.* Berlin: Springer.

S, J., & Alagarsamy, K. (2006). An evolutionary model for the management of software projects. *International Conference on Global Manufacturing and Innovation*, Coimbatore, India.

Schmidt, A. (2005). *Bridging the gap between knowledge management and e-learning with context-aware corporate learning.* Lecture Notes in Artificial Intelligence. Berlin, Heidelberg, New York: Springer-Verlag.

Smith, D. L. (1995). Effective teaching without lecturing. *ASEE Annual Conference Proceedings*.

Stevens, W. R. (1999). *Advanced programming in the UNIX environment*. Addison Wesley.

Stevens, W. R., Fenner, B, & Rudoff, A. (2003a). *Unix network programming, The sockets networking, API* (3rd ed., Vol. 1). Pearson Education.

Stevens, W. R., Fenner, B, & Rudoff, A. (2003b). *Unix Network Programming, Interprocess communication*, *API* (3rd ed., Vol. 2). Pearson Education.

Tatsuoka, M. M. (1998). *Multivariate analysis: Techniques for educational and psychological research.* Macmillam Publishing.

Vestal, W., (2005). Integrating knowledge management and organizational learning. American Productivity & Quality Center. http://www.apqc.org/portal/apqc/site/content?docid=119786.

Biographies

Justus S is a doctoral candidate at Madurai Kamaraj University, Madurai, India. His research areas are Knowledge management, software process improvement, ORDBMS and software metrics. He has several international publications including IEEE and ACM. He has visited University of Potsdam, Germany for research paper presentation and is visiting France and Israel during this year to present his research works in IEEE International Conferences.

He is a member of review committee for IJSEKE and Journal of IET, USA. He is also serving as senior member of faculty in the department of computer applications, K.L.N. College of Information Technology, India.

Dr. S.K. Ramalingam, M.Sc, PhD., retired from Madurai Kamaraj University, Faculty Department, as Senior Professor, Head & Coordinator and Coordinator, Center for Special Assistance. He has extensively researched for 25 years and has authored more than 70 research papers and 3 reviews in International Journals. He has been awarded medals and certificates for outstanding contribution to Chemical services by American Chemical Society and a medal by British Education Journal for research work in Teaching techniques. Presently, he is Dean (academic) in K.L.N. College of Information Technology, India.