Merging Operating Systems and Computer Networks: Why and How

Aleardo Manacero Jr.

Computer Science and Statistics Dept.
Paulista State University - UNESP
São José do Rio Preto, Brazil, 15054-00

Abstract

This paper describes the innovative approach of integrating the operating systems and computer networks courses into a single, one semester, course for computer engineering and/or computer science students. This merge is justified by the today's intrinsic connection between these contents, which is tightening during the 90's by the strengthening observed with distributed operating systems implementations. Some of the relevant aspects that lead to this merge are fully described just before the explanation on how both contents can be taught at once. This approach has been tested at UNESP and the preliminary results are shown here.

1 Introduction

Current curricula in most of the computer engineering and computer science schools try to expose all students to a large amount of knowledge. This knowledge is seldom used after the graduation, during the students professional career. While this approach may be backed by arguments such as *"looking for a thorough and deep understanding of the whole science"* or *"there is no benefits in an early specialization"* or even *"the lack of any information reduces their chances while competing for a position"*, it is also easy to criticize it by arguments such as *"it is impossible to know many things about many subjects"* or *"industries always look for specialists"* or *"there is no time to receive every piece of information"*.

The excess of information results in an excess of time spent inside the classrooms, what is not even close a good measure of proper learning process. In specific with computer networks and operating systems, the time needed to get a sound formation is very long, demanding many practical works. Many schools adopt the methodology of teaching them separately, which is called conventional hereafter. However it is a very well known fact that operating systems (OS) courses deal with complex topics and is feared by almost every student, and that computer networks (CN) may be approached exactly the same way. As a result of this, these courses are not only time consuming but also boring to many students since they cannot see any of the actual relationships between them.

From this point of view, one may try to tie both topics into a single topic that is much more interesting. Studying operating systems with computer networks in mind, or the other way around as well, is more meaningful than studying them under different views. Such course is not time consuming and enables a complete understanding of the technological issues in these fields.

This paper describes an integrated approach for teaching operating systems and computer networks at once. In the following sections the reader will find first a brief description of how these topics are taught under the conventional approach and the reasons why it should be reviewed. Then, it is presented the main aspects that have to be considered while planning an integrated course, followed by an example of such course. Preliminary results achieved with almost 60 students are shown and discussed during section 4. From these results we draw some conclusions about the feasibility of the integrated approach as well as indicate some adjustments to continue with its application.

2 Why integrate OS and CN courses

Before describing why one should integrate operating systems and computer networks courses it is necessary to review how these courses have been taught and the problems and advantages of such approach. From this review one can devise the adequacy of integrating the body of knowledge of both fields. In this section the reader will see first a brief description of the conventional approach and the reasons why this new approach should be considered.

2.1 The conventional approach

The usual approach in computer engineering and/or computer science undergraduate schools is to give separate courses on computer networks and operating systems, mirroring the textbooks in this area and the vertical structure of conventional education. Therefore, there are one or two semesters courses on operating systems and, usually, one more semester on computer networks.

Operating systems courses are strongly oriented to OS implementation, covering topics on concurrent programming, processes, I/O and memory management, file systems and, if the time is enough, distributed operating systems [2, 3, 5, 6]. These topics mean a large amount of knowledge, which has to be quickly managed by the students. The problem is that that knowledge is also complex if the course is oriented to the OS design and analysis (as it is in most of the cases). This leads to its extension over two semesters, the first covering most of the theoretical knowledge and the second one covering design issues and practice.

On the other side, computer networks courses are not oriented to implementation, covering usually protocol definitions and the description of a few real implementations, such as TCP/IP and ATM [4, 7]. Therefore, practicing with CNs is not a major concern during the course, except when a second course is introduced in the curriculum.

Then, it is easy to verify that on both courses there is a large amount of theoretical knowledge. Practical work is performed at the expense of more time dedicated to theses topics, what usually means an overloading in the undergraduate majors. By the way, most of undergraduate majors on computer engineering/science are already overloaded, what conflicts with the desire that the students have more free time to practice the topics taught at classroom. Another aspect that deserves some consideration is that not every student intends to actually design an operating system or a network. Then, it is not a good choice to turn these design courses as mandatory ones. These problems were the main reason to integrate both courses into a single, one semester, mandatory, leaving the design issues as optional courses for those who are interested in them. A thorough description of these and other reasons is given in the next section.

2.2 The integrated approach

The main question here is why integrate these courses. The visible answers are because students cannot spend too much time in their classrooms or because they have a few similarities or because they may be devised as a correlated. Each one of these answers has, indeed, its degree of truth and has some implication over the other ones. From a historical point of view, the decision about their integration was made first from the students' time aspect, then from the similarities and correlations among OS and CN.

Therefore, the most important argument against the conventional approach is that it is more convenient to leave students with much more time to practice those topics that spring their interest. This was the starting point for the establishment of the computer science major's new curriculum at UNESP, campus of Rio Preto. Therefore the curriculum's main goal is to give a strong basis to the students while avoiding the course overloading. This option demands a reduction in the number of courses that are mandatory for every student and, consequently, implies in the search for courses that can be merged or eventually eliminated.

As a consequence of this orientation, every course merge had to be tested and analyzed. The good choices for a merge involved those courses that have similarities and some degree of interference between their contents and, mainly, could be used as introductory courses to specific areas of computer science. Among them the merge of operating systems and computer networks was chosen because this mixture leads to the area of software/hardware interfacing for both stand-alone and integrated environments.

The similarities among computer networks and operating systems fall mainly into the fact that both softwares are used to control hardware and to provide a virtual environment to the end user. This fact can be used to drawn common issues on their specification and implementation, making their contents integration easier.

The integration between the contents is the key aspect in this approach. It is feasible through the determination of the common issues around them, such as the synchronization of concurrent processes for OS's and the messages flowing between layers for CN's.

Concluding this section, the following list summarizes the main arguments favorable to an integrated approach to teach computer networks and operating systems:

- Both courses are too dense to be fully taught to students not interested in working in those fields;
- Practicing over the topics of study is much more productive than listening to theories in classrooms;
- OS and CN are similars in the sense that both are used to provide a virtual environment to end users;
- There are several common issues on their specification and implementation.

3 How to integrate OS and CN courses

Besides their similarities in use and some internal issues, the integration of both topics into a single course is not a commonplace. A student has to perceive the differences between them in order to understand both as single systems that may eventually be linked. At UNESP the way to provide such view is to show initially an application that needs the interaction between operating systems and computer networks. One example for such application is high performance computing (HPC) using distributed environments.

Using HPC as the starting point is very practical since nowadays it can be performed through personal computers linked by a simple network. Avoiding deeper concepts of parallelization, the student is leaded throughout many examples of problems that require the use of HPC and can be parallelized using PVM [1] among other tools. The availability of PVM as an inexpensive method to achieve high performance triggers the perception of interaction between OS and CN.

After the completion of this phase, the course changes its direction, going to specific details of computer networks and operating systems. Each of these details should not be seen as isolated issues to not go back to the conventional approach. Instead of this, after a brief description of why and how a network protocol is defined in layers, the students are introduced to the concepts of concurrent programming and tasking. This establishes the need for packet synchronization and mutual exclusion over the physical devices along the net. Afterwards, these concepts are pushed into process management inside operating systems.

This process goes back and forth until each topic, of both original courses, could be taught. Unfortunately some of them cannot be linked explicitly with topics of the other course and have to be picked when some related topic, from the same original course, is approached. One of such topics is the service of catalog management (directories) by the file system's section of an operating system, which may be approached when networks security leads to the file systems security services and, therefore, to other services provided by the file system.

By the end of the semester every student should be able to, not only recognize the concepts involved with every topic of operating systems and computer networks, as isolated pieces of information, but also to recognize the possible interactions between them and the similarities of problems and solutions to achieve their goals. At this point it is not expected from them to be experts, not even to have a sound knowledge, in any of these fields. This expertise, or sound knowledge, will be achieved only by those who want to work in areas that relate either with computer networks or operating systems.

Although the professor in charge of this course may require some small design works, full design works are left to the later, in depth, courses. These courses are taken only by the students who are really interested in having a stronger preparation in these areas. Since the audience of these later courses is more motivated, it is possible to go further in the theory and demand better design works than it is possible with the conventional approach.

3.1 A suggested series of lectures

The integrated approach presented here was tested with computer science students of UNESP at Rio Preto. In this section it will be presented a sketch of a series of lectures that cover the scheduled topics. Although only the most meaningful lectures are reviewed, they provide enough information to understand the principles used to achieve success with this approach.

Before starting with the lectures' sketch, it is needed to describe the environment where the experience took place. First, this new course was created under the philosophical principles established with the new curriculum for the computer science major. Besides it is scheduled for the fourth semester, last year its attendance included students in their sophomore and junior years. This mixture occurred because the necessary adaptations that junior students, who did opt for the new curriculum, had to perform in order to graduate under the new curriculum.

The background expected from the students is the one typical for a sophomore student at UNESP. This background includes, by the end of the third semester, the concepts of structured programming (in three 60 hours' courses) covering fundamental programming and data structures. They also went through one semester of digital circuits, including both theoretical and practical classes. They have fundamental knowledge about automata theory and compiling too. This somewhat large background permits a fast development of all topics during the semester.

Figure 1 shows the lecture's sketch. There, the vertical positioning indicates when each topic is approached. The number of lectures spent in each topic usually ranges from one to six, each one with 100 minutes of duration. The distribution of lectures along the central topics (OS and CN) is also easy to be figured out, as well as the connections made to flow along them.

These lectures provided an insightful view of both topics, bringing the attention of all students over the fact that only the integrated use of both could enable the achievement of high performance computing at a low cost. Their knowledge about these topics is not as sound as the knowledge achieved with two or three semesters covering the same subject. However, it is large enough to let them to choose this field of work if they feel comfortable with it or any other field if they do not. Since not everyone will be an operating systems or computer networks designer/manager, this small initial knowledge is enough and practical.

4 Comments about its application

Although the experiments with the integrated approach still very incipient, there are some results that may show how the students received such modifications and how their expectations improved within the semester. The experiment performed until now included only one class, with 59 students enrolled on it. As mentioned before these students were in their sophomore and junior years, what let some space to heterogeneousity on their background.

During the semester the students answered some pool questions to quantify the experiment evolution. These pools should be analyzed differently for sophomore and junior students, but this was not done to preserve the anonymity of those who answered the questions. However, from informal talks was possible to verify that the results from the pools would have very little variances if accounted separated for each group. This indicates that both groups had similar impressions about the course.

Among all results from the pools, the most relevant ones are presented at table 1. One may see from there that the average grades got better as the semester went on. This evolution would be expected since none of the students had a previous knowledge about what was going to be taught during the course, exception made to the historical preconceptions that operating

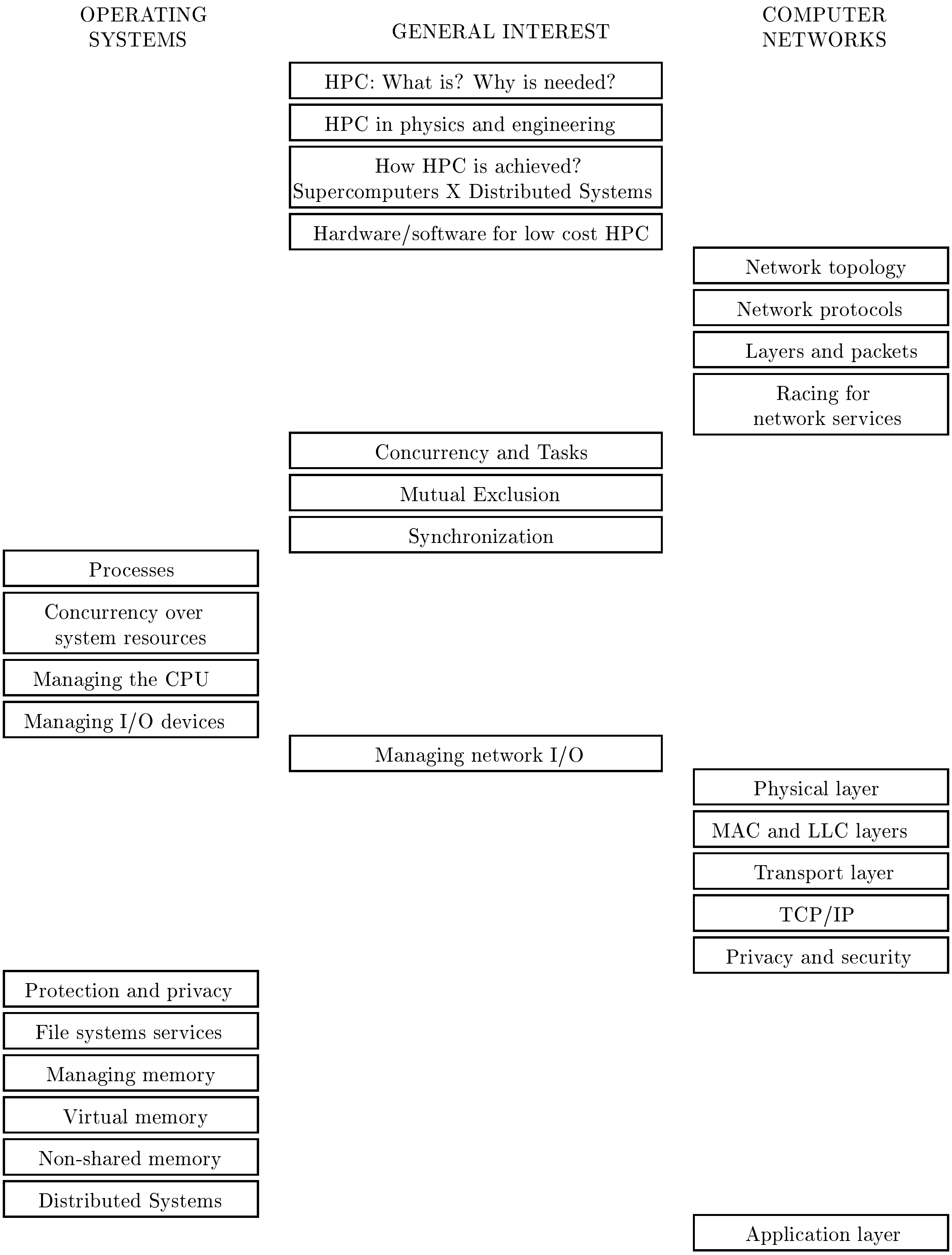

Figure 1. Sketch of a series of lectures for the integrated approach.

Table 1. Relevant results from the course pools.

	Average grade (1-5, the higher the better)		
Question	first week	2 months	last week
Do you know what are the aimed goals?	2.0	3.8	4.5
Are you stimulated about studying this?	3.2	3.2	4.0
How much do you like OS?	2.0	2.8	3.8
How much do you like CN?	3.2	3.6	4.3
Is this course better than two separate courses?	—	3.6	4.5

systems is tough and computer networks have a boring introduction (the protocols).

Therefore it is easy to explain why the students' expectations changed during the semester. Since this was the first time that this course was offered, they could not expect anything different from the conventional approaches. During the course development they could devise a new perspective for both, operating systems and computer networks, seeing their relationship and how similar problems have similar solutions besides they are applied to different systems.

5 Conclusions

After the previous section the reader may draw the same conclusions that this author had come to. The students show a better understanding of why they study operating systems, leading to not only better grades in their evaluations, but also to a more pleasant study of its internals. On the other side, almost all students have good expectations from computer networks, mostly due to internet and the massive merchandising of CN as the solution of all computing problems. Although this expectation is a little bit erroneous, they were less disappointed than older students that took the conventional course.

Even these results are enough to indicate that the approach is feasible and should be evaluated by other schools. From this experience it is easy to tell that the integrated approach is applicable even in a more conventional curriculum, where the standard OS and CN courses still in place. For these cases, the difference would be that all courses are mandatory instead of electives as they are in the UNESP case. The curriculum load will not change when the mandatory courses still in place but the students' interest in these topics would change for better.

Since everything has some imperfections, there were some negative aspects raised by the students and that should be reviewed for the next offerings. The major problem raised was that the evaluations were too hard when compared to the volume of information that they have to absorb. This is indeed a typical students' complaint but it has to be better weighted since this was the first time that the course was offered and its scales came from the conventional approach.

A second complaint was about the lack of practical works. Although this course was originally conceived as a plain theoretical course, it may include some small design works or, on the other way around, some research works. Now the author has the feeling that the students' stimulation may rise if they are challenged with practical problems and/or deeper theoretical questionings.

Putting everything into a single scale the experience was very successful. This implies that the integrated approach to teach OS and CN as correlated subjects is attractive and brings the students' interest about each subject to a higher level. The preliminary results achieved in 1997 should be improved in the next few years since the problems detected are under analysis in order to provide a better course.

References

[1] Geist, A., Beguelin, A., Dongarra, J., Jiang, W. Manchek, R. and Sunderam, V.; "PVM 3 Users's Guide and Reference Manual"; ORNL/TM-12187, September 1994.

[2] Deitel, H.M.; "Operating Systems - 2nd Ed.", Addison-Wesley, 1990.

[3] Peterson, J. , Silberschatz, A. and Galvin, P.; "Operating Systems Concepts - 3rd Ed.", Addison-Wesley, 1991.

[4] Stallings, W.; "Local & Metropolitan Area Networks - 5th Ed.", Prentice-Hall, 1997.

[5] Tanenbaum, A.S.; "Modern Operating Systems", Prentice-Hall, 1992.

[6] Tanenbaum, A.S. and Woodhull, A.S.; "Operating Systems: Design and Implementation - 2nd Ed.", Prentice-Hall, 1997.

[7] Tanenbaum, A.S.; "Computer Networks - 3rd Ed.", Prentice-Hall, 1997.