https://doi.org/10.15414/isd2018.s10.03

THE REDESIGN OF THE SEMINARS IN AN OPERATING SYSTEMS COURSE

Ján Genči[1], Sandra Antoliková[2]
Technical University of Košice[1,2]
Department of Computers and Informatics
Department of Languages
Letná 9, Košice, Slovak Republic
e-mail[1,2]: genci@tuke.sk, Sandra.Antolikova@tuke.sk

Abstract

The paper discusses some approaches we have applied in our department to overcome some current problems identified in our higher education system. During the last few years, we have developed a detailed study manual which guides students during their preparation for seminars. The study process is supported by formative tests and assignments for self-preparation. Two summative knowledge term tests and a comprehensive term assignment are also presented. Moreover, the summative and formative knowledge tests are targeted at higher levels of taxonomies of educational objectives in the cognitive domain. The majority of activities are supported by LMS Moodle and custom system for evaluation of term assignments. All course activities are tied together by a complex scoring system. We believe that the identified problems are common to the majority of higher educational institutions in our region, and our solutions can serve as inspiration for other academics.

Keywords: *operating system course, knowledge assessment, formative and summative assessment, course redesign, course assignment*

JEL Classification: *I21, I23, Y9*

1 Introduction

Massification of higher education systems in the Central European region and openness of educational systems of European countries for students from our country cause many new problems in Slovak higher education which we did not face in the past. Most of the problems are problems which were solved in the past by Western universities up to 40 or 50 years ago; it sometimes seems that it is enough to get inspiration from Western universities by looking at how they once

solved the problems we now face. However, academics from Western universities report that their universities are faced with different kinds of problems [5],[7],[8], so it is not so easy to rely on their experience. The problem which arises most commonly is an increase in the number of universities (about 40 at this moment), as well as the number of students. In spite of the fact that our department, the Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice, has not faced a massive increase in the number of students, the consequences of the massification of higher education are present in all their complexity – primarily in a decrease in the students' interest in studying. In addition, as we have already hinted at above, another problem is the relatively large number of Slovak students studying abroad (mainly in the Czech Republic). The next problem of regional universities (the majority of Slovak universities, including ours) is that quite a high number of capable students from our region study in the capital city at top Slovak universities.

2 Common reasons for a course redesign

We are not alone in thinking about current problems in Slovak higher education and in trying to experiment with various solutions ([1-3], [9-10]). However, we think that we have been able to find some common reasons which form the basis of our problems. Our thoughts stem from factors associated with intelligence presented by Thurston. In [4] the author states the following factors:

- the ability to affect verbal relationships,
- word fluency,
- the ability to manipulate relations in the spatial dimension,
- perceptual abilities,
- the ability to manipulate numbers
- memory,
- general ability to reason,
- the ability to reason inductively,
- the ability to reason deductively.

We assume that (the level of) intelligence across the population can be modeled by normal distribution and that university students are mentally the most capable part of the population. In this context we deduce that the massification of university education has lowered the average level of the above factors associated with intelligence.

In [6] (based on work [11]) the authors state that universities in North America underwent the process of massification about a hundred years ago while

Western Europe universities started this process 50-60 years ago. According to [6] during the transformation of the former educational system (authors refers to this system as elite with 15 % of high school graduates entering higher education) to the massive form of education (50% of high school graduates entering higher education), the application of old managerial methods can be still observed. We claim that the same might be observed in the educational process itself – we still try to apply approaches (ways of teaching) we used in the past during the era of the elite form of education.

We can take into consideration one more factor related to intelligence - intrinsic motivation to acquire new knowledge. We can reformulate it in the context of motivation to study – the lower the level of intelligence, the lower the level of motivation to study. This could explain the frustration of teachers who teach in current university classes. On the other hand, students who have been allowed to enter a university, expect to get appropriate knowledge, skills and training. They are not "guilty" of not meeting the expectations of ordinary teachers. It is up to us, teachers, to adjust educational processes in order to be able to fulfill students' expectations.

Software engineering, similar to computer science and even mathematics, is a purely abstract discipline dealing with data, information and the ways of its processing. In this context, the factors of intelligence, such as the ability to affect verbal relationships, the ability to manipulate relations in spatial dimension, memory, the general ability to reason, the ability to reason inductively and the ability to reason deductively are all crucial for students who study information technologies. A decrease in these abilities consequently influences the educational process. In this case, the main task of a teacher is to try to eliminate such a decrease in students' abilities, and to help less skilled students to increase their productivity.

3 Course organization in the past

In the past, in the period of elite education, the educational process (lectures and seminars) was rather loosely organized. The main emphasis was on lectures – the main educational purpose was to transfer teachers' knowledge to students (teacher-centered education). The seminars were usually based on the lecture contents and gave students the opportunity to understand presented topics deeply or to develop certain mental skills (e.g. in mathematics - calculation of limits, derivations, or integrals).

It was assumed that students would participate in the educational process regularly on a day-to-day basis, were able to understand the presented material, could remember facts and were able to derive reciprocal relationships. Nowadays,

complaints about students' attitudes to studying, coming from all sides, do not reflect the change, but instead hark back nostalgically to the era of elite higher education (elite - as defined in [6]).

In our subject, Operating Systems, lectures and seminars curricula were separated about 20 years ago. The main goal in doing this was to ensure that students have the opportunity to acquire required skills in the field of system programming in UNIX/ Linux operating systems and not just to memorize the basic principles.

4 Redesigned parts of course

Based on the factors associated with intelligence, it can be assumed that the average student's ability in mental processing of the lecture contents, the ability to work with literature, the ability to derive relationships between presented facts etc. is less now than in the past (during the elite university study era). To eliminate these drawbacks, a supportive study guide has been developed to help our students to cope with necessary topics. Each chapter of the study guide (published as PDF files) is introduced in the form of a mind map presenting essential concepts for a given topic (Figure 1). The objectives, the motivating scenario and the estimated time for completion of the presented topics (Figure 2) have been specified for each topic and its sections. A student is guided step by step through the study material (Figure 3) with explanation of key concepts. Moreover, the students are required to test their understanding of the topic with presented practical examples.

As mentioned above, it can be assumed that teachers deal with students with lower intrinsic motivation to study than most students had in the past. It means that we have to provide external motivation for students to study. Formative knowledge tests are used for these reasons Figure 4). These have been developed for each seminar topic in our Operating System course. The goal of each test is to motivate students to prepare for the following seminar. Sets of home assignments for each topic have also been developed to motivate students to work after seminars, We also provide opportunity for students to practice acquired knowledge and practical skills in the UNIX/Linux environment. Currently, students upload their home assignments to the LMS Moodle, where the teacher checks them manually.

Besides providing students with study guidance and formative knowledge tests, systematic verification of knowledge is also very important. Two summative knowledge tests have been developed to objectively check knowledge and intellectual abilities acquired by students. The aim was to develop test items classified at the higher levels of Bloom's taxonomy (Figure 5, 6). The higher level items

ensure that the student is able to demonstrate understanding (not memorizing) of the required knowledge and intellectual skills as well.

Figure 1 **Study guide – topic mind map**

Figure 2 **Study guide – objectives, estimated time and scenario**

Topic: OS Unix file manipulation		
Keywords	OS UNIX/Linux file system, files, file manipulation, inode	
Aims	Remembering	of basic system calls for file manipulation
	Understanding	of system call parameters and the connection between them
	Application	of system calls for: • file opening, reading and writing • getting file metadata • setting access rights • file deletion
	To know	how to apply acquired knowledge to create relevant programs
Estimated time	105 min.	
Scenario	Sofia already knows how to work with main pages, and she is also able to intercept mistaken system calls. Now she needs basic knowledge of system calls relevant to file manipulation.	

The home assignments and the summative knowledge tests concentrate on the particular topics presented during the seminars. The comprehensive term assignment (Figure 7) was created to give students opportunity to connect together almost all the knowledge and intellectual skills presented throughout the course into one common solution. The goal is to verify knowledge and practical skills in the area of UNIX/Linux system programming, covering most of the topics presented during seminars – file handling, processes creation, program execution, communication between processes (pipes, signals, shared memory), synchronization, and network communication (UDP and TCP protocols ,IP protocol). Moreover, in order to increase the productivity of teachers, a web - based application has been developed allowing students to upload their assignments. The uploaded assignment is compiled and tested. In addition, every student is provided with the detailed protocol of translation and testing. The number of uploads is unlimited, therefore, the student has chance to modify his or her assignment. The same assignment is used for all students so the application checks all uploaded solutions to the problem of plagiarism. The teacher has an overview of the state of the student's assignment (submitted, not submitted, number of attempts, date of their successful submission, level of originality or plagiarism, etc.) and the possibility to compare the solutions of students who may be suspected of having plagiarized (Figure 8).

Figure 3 **Study guide – step by step instruction**

Actions

STEP 1 – to learn syntax and semantics of system calls for input/output:
All inputs and outputs are performed by the system calls `read()` and `write()`, respectively:

Syntax:

```
#include <unistd.h>
read(int fd, char *buf, size_t count);
write(int fd, const char *buf, size_t count);
```

Semantics:

- `read()` reads *count* bytes from the file descriptor *fd* to the buffer *buf* and returns the number of bytes read or 0, when the end of the file is reached, or -1 on an error.
- `write()` writes *count* bytes to the file descriptor *fd* from the buffer *buf* and returns the number of bytes written or -1 on an error.

Figure 4 **Formative knowledge test item**

Fill in: "A semaphore is an integer whose value is never allowed ... zero." (see man 7 sem_overview, or http://www.kernel.org/doc/man-pages/online/pages/man7/sem_overview.7.html)

Answer:

As mentioned above, the problem with plagiarism was foreseen in the case of term assignment, therefore, the application was developed with plagiarism detection functionality from its beginning. Unexpectedly, certain problems with the formative tests had to be solved because of plagiarism. These tests are designed to guide a student through study materials, and thus, have a fixed structure. Some students decided to share their item answers on social networks and many of their colleagues just copied their answers and used them in their own tests. In the first stage of the problem solving it was quite easy and quick to identify not only the students who copied the answers, but also those who shared the correct answers on social networks by comparing their answers (because of open ended items and multiple valid answers, e.g. using or not using diacritics, not answered items, etc.). In the next phase, the time stamping item, provided by LMS Moodle, was used for more complex analysis and identification of the suspected students (Figure 9, 10).

Figure 5 **Summative test item – files**

Consider a file qwe.txt, the current status of which (content and the position of the file pointer) is, at the time of execution of the system call read (), defined as follows:

```
Characters:     bkkyxamtwifeiphqirkqyyvkry
position:       0000000000111111111122222
                0123456789012345678901234 5
File offset:    --------------------^
```

and the fragment of code:

```
//--------- beginning of the fragment
int fd,i;
char buffer[80]="PWNDSTY";
 //             0123456
...
fd=open("qwe.txt",O_RDWR);
...
i=read(fd,&buffer[1],2);
  ...
//--------- end of the fragment
```

Specify the content of the string variable **buffer** after exectution of the above fragment of coce in the C language (Specify the string exactly, taking into account the letter case. The remaining position of the variable **buffer** contain a binary value of zero).

Answer:

Figure 6 **Summative test item – processes**

Consider three programs with the source code presented in the following table:

Program p1	Program p2	Program p3

`main(int arc,`
`char *argv[],`
`char **env){`
`putchar("x");`
`execve("p1",argv,env);`
`}`
	`main(int arc,`
	`char *argv[],`
	`char **env){`
	`putchar("y");`
	`execve("p2",argv,env);`
	`}`
		`main(int arc,`
		`char *argv[],`
		`char **env){`
		`putchar("z");`
		`execve("p3",argv,env);`
		`}`

Specify the first four characters printed on `stdout` which will be produced by this set of programs if we start execution by program `p3`.

Answer:

The strict scoring system (Table 1) was designed to ensure objectivity of term assessment. Students gain points for any given activity (such as formative test, homework, summative test, assignment, documentation) At the end of the semester, each student gets points based on the sum of all scored activities and the teacher's subjective evaluation, which represents less than 20% of all the points awarded to assess a student's activity.

Figure 7 **Comprehensive assignment**

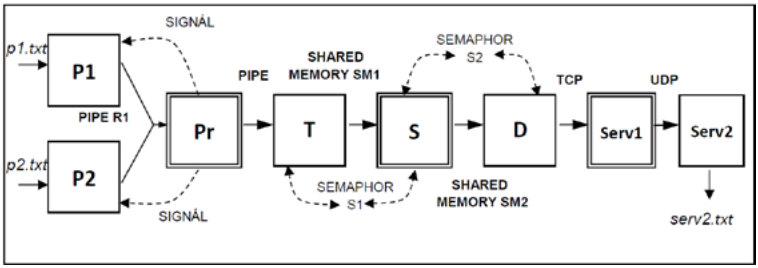

Figure 8 **Computer application for assignment evaluation**

ID	Surname	FirstName	Group	Assignment	State	Submission order	Originality	Report	More options	
1234	[redacted]	[redacted]	54	OS unix	Accepted, 15. 01. 2016 15:48:56, 70 # of submissions:1		100%	Not submitted	Assignment	Evaluation
59293	[redacted]	Peter	52	OS unix	Submitted, not accepted	-	0%	Not submitted	Assignment	Evaluation
64463	[redacted]	Eduard	32	OS unix	Accepted, 18. 01. 2016 00:42:59, 72 # of submissions:23		100%	Submitted	Assignment	Evaluation
65336	[redacted]	Richard	11	OS unix	Accepted, 12. 01. 2016 13:12:15, 42 # of submissions:3		63%	Submitted	Assignment	Evaluation
65608	[redacted]	Jozef	52	OS unix	Accepted, 09. 01. 2016 18:16:29, 27 # of submissions:44		100%	Submitted	Assignment	Evaluation
65646	[redacted]	Kristian	11	OS unix	Submitted, not accepted	-	0%	Not submitted	Assignment	Evaluation
65649	[redacted]	Eugen	52	OS unix	Accepted, 11. 01. 2016 17:43:00, 37 # of submissions:8		100%	Submitted	Assignment	Evaluation
65670	[redacted]	Daniel	52	OS unix	Accepted, 12. 01. 2016 07:18:30, 39 # of submissions:7		100%	Submitted	Assignment	Evaluation
65679	[redacted]	Miroslav	11	OS unix	Submitted, not accepted	-	0%	Not submitted	Assignment	Evaluation
65689	[redacted]	Rastislav	52	OS unix	Not submitted yet	-	0%	Not submitted	Assignment	Evaluation
65730	[redacted]	David	52	OS unix	Not submitted yet	-	0%	Not submitted	Assignment	Evaluation
65779	[redacted]	Martin	52	OS unix	Not submitted yet	-	0%	Not submitted	Assignment	Evaluation
65806	[redacted]	Jan	52	OS unix	Accepted, 14. 01. 2016 10:32:08, 63 # of submissions:1		66%	Submitted	Assignment	Evaluation
65835	[redacted]	Maros	52	OS unix	Accepted, 08. 01. 2016 08:04:06, 23 # of submissions:6		74%	Submitted	Assignment	Evaluation

Figure 9 **Formative test – evaluation of plagiarism – data**

F	G	H	I	J	K	L
CategQ	NameQ	TMPSTM	Attem	Duration	TM delta	TM diff.
12-Subory-porozume	Post-Q01-001	1361911487	1	42594	728	32
12-Subory-porozume	Post-Q01-002	1361911893	1	42594	1134	406
12-Subory-porozume	Post-Q01-002	1361912791	1	42594	2032	898
12-Subory-porozume	Post-Q01-002	1361952259	1	42594	41500	39468
12-Subory-porozume	ost-Q01-003-F	1361952259	1	42594	41500	0
12-Subory-porozume	ost-Q01-004-F	1361952259	1	42594	41500	0
12-Subory-porozume	Post-Q03_001	1361952259	1	42594	41500	0
12-Subory-porozume	ost-Q04-1_1(	1361952259	1	42594	41500	0
12-Subory-porozume	ost-Q09-1_0(	1361952259	1	42594	41500	0
12-Subory-porozume	ost-Q10-V2-0(	1361952259	1	42594	41500	0
Week-02-Subory-II	FL2-01	1361952259	1	42594	41500	0
Week-02-Subory-II	FL2-02	1361952259	1	42594	41500	0
Week-02-Subory-II	FL2-03	1361952318	1	42594	41559	59
Week-02-Subory-II	FL2-04	1361952318	1	42594	41559	0
Week-02-Subory-II	FL2-05	1361952318	1	42594	41559	0
Week-02-Subory-II	FL2-06	1361952399	1	42594	41640	81
Week-02-Subory-II	FL2-07	1361952399	1	42594	41640	0
Week-02-Subory-II	FL2-08	1361952399	1	42594	41640	0
Week-02-Subory-II	FL2-09	1361952399	1	42594	41640	0
Week-02-Subory-II	FL2-10	1361952399	1	42594	41640	0
Week-02-Subory-II	FL2-11	1361952399	1	42594	41640	0
Week-02-Subory-II	FL2-12	1361952399	1	42594	41640	0
Week-02-Subory-II	FL2-13	1361952470	1	42594	41711	71

Figure 10 **Formative test – evaluation of plagiarism**

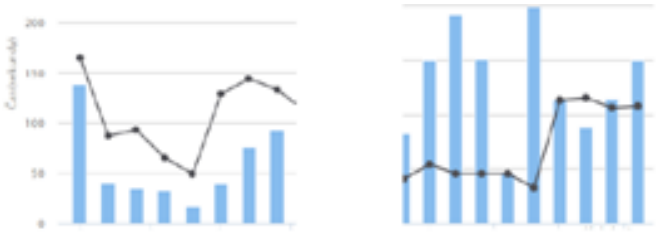

Table 1 **Course scoring system**

Required activities	Formative tests	Home assignments
Summative test I. **10 points**	2 points for each test over 75% (8 weeks)	2 points for each series of home assignment (9 weeks)
Summative test II. **10 points**		
Term assignment **20 points for program** **10 points for documentation**		

5 Conclusion

The presented activities have been developed and introduced step by step over a period of several years. The complex approach described here, has been implemented during the last two years. From the teacher's point of view, we were really able to implement and refine all presented activities. The questionnaire was used to gather students' opinions. The responses vary. Some students were not satisfied with the requirement to prepare systematically, while others were satisfied and appreciated the fact that they were well prepared for midterm and final tests. A small number of students failed to gain enough points during the term and gave up before its end. We assume that it could be a future challenge for us to identify such students as soon as possible and provide them with the support they need to cope with their study problems.

Acknowledgements

This article was published with the support of EUNIS-SK

References

1. JELEMENSKA, K. (2012). Integration of ICT and Unconventional Teaching Approaches into the Digital Systems Design Education Towards its Efficiency Enhancement. I*CETA 2012 - 10th IEEE International Conference on Emerging eLearning Technologies and Applications, Proceedings*, pp. 173-178.
2. JELEMENSKÁ, K., KOINE, P., ČIČÁK, P. (2013), Enhanced Classroom Presenter. In: *Sobh, Tarek & Elleithy, Khaled: Emerging Trends in Computing, Informatics, Systems Sciences, and Engineering, Lecture Notes in Electrical Engineering 151, Springer*, pp. 973-983.
3. KOREČKO, Š., SORÁD, J., DUDLÁKOVÁ, Z., SOBOTA, B., (2014), A Toolset For Support of Teaching Formal Software Development. *Lecture Notes in Computer Science - LNCS 8702, Springer*, pp. 278-283.
4. KUBANI, V. (2010). General psychology. *University of Prešov*, 2010. ISBN 978-80-555-0172-7 (In Slovak).
5. NEUBAUER, D. (2014). The University in the Context of Continuing Globalization.*In: Knowledge Studies in Higher Education. Vol. 1, Springer*, pp. 29-44.
6. PRUDKÝ, L., PABIAN, P., ŠIMA, K. (2010). Czech Higher Education: On the Path from Elite to Universal Education 1989-2009. *Grada Publishing a.s., 162 pages (in Czech)*.
7. SHIN, J. CH. (2014) The University as an Institution of Higher Learning: Evolution or Devo-lution? *In: Knowledge Studies in Higher Education. Vol. 1, Springer*, pp. 13-28.
8. SHIN, J. Ch., TEICHLER, U. (2014). Future of the Post-Massified University at the Crossroads: Restructuring Systems and Functions. *In: Knowledge Studies in Higher Edu-cation. Vol. 1, Springer*, pp. 1-12.
9. ŠIMOŇÁK, S. (2014). Using Algorithm Visualizations in Computer Science Education. *Central Eu-ropeanJournal of Computer Science, September 2014, Volume 4, Issue 3*, pp. 183-190.
10. TELEPOVSKA, H., TOTH, M. (2013). Document Support of Relational Algebra Knowledge Assessment. *In: Sobh, Tarek & Elleithy, Khaled: Emerging Trends in Computing, Informatics, Systems Sciences, and Engineering. Lecture Notes in Electrical Engineering 151, Springer*, pp. 475-485
11. TROW, M. (1973), Problems in the Transition from Elite to Mass Higher Education. *Carnegie Commission on Higher Education. Berkeley, California.*