*Journal of Engineering Education Transformations,*
*Volume 30, No. 4, April 2017, ISSN 2349-2473, eISSN 2394-1707*

# FBL: An Activity to Improve Students' Conceptual Understanding of the Course Operating System

**Sujit N.Deshpande[1], Sunita M. Dol[2]**
[1]Computer Science and Engineering Department, Walchand Institute of Technology, Solapur, India
[2]Computer Science and Engineering Department, Walchand Institute of Technology, Solapur, India
[1]sujit.sujitdeshpande@gmail.com
[2]sunita_aher@yahoo.com

**Abstract:** Free and open-source software (FOSS) is computer software which is freely licensed software to use, copy, study, and change the software in any way. FOSS can also be used for teaching the course. We have designed an activity FOSS Based Learning (FBL). In FBL, instructor uses the free open source software to teach the topic and problem solving based on that topic. In this work, CPU-OS Simulator as FOSS is used. This tool can be used for conceptual understanding of various topics of Operating System, Compiler Construction, System Programming and Computer Architecture. In this particular study, FBL is used for studying and evaluating the performance of scheduling algorithms in Operating System course for Computer Science and Engineering students. The results of teaching this topic using FBL is presented in this paper. For this study, one group pre-post test method is considered with three learning objectives (LOs) (1) To draw Gantt chart for four scheduling algorithms(LO1) (2) To calculate waiting time for individual process and average waiting time, for four scheduling algorithms (LO2) and (3) To calculate average turnaround time for four scheduling algorithms (LO3). Also the students' feedback about this FBL is also given in this paper.

**Sujit N.Deshpande**
Computer Science and Engineering Department,
Walchand Institute of Technology, Solapur, India
sujit.sujitdeshpande@gmail.com

**Keywords:** Free Open Source Software (FOSS), FBL(FOSS based Learning), Likrerts' Scale, t-Test, CPU-OS, Visualization, Operating System (OS).

## 1. Introduction

Free open source software (FOSS) means the freedom to run the program, for any purpose, the freedom to study how the program works, and adapt it to your needs, the freedom to redistribute copies, the freedom to improve the program (https://en.opensuse.org/Free_and_Open_Source_Software). Simulation can be used in teaching process to enhance students' conceptual understanding (Rutten, N. et. al., 2012). It is also used to practise and solve problem in realistic environment (Van Berkum et. al., 1991). Students can change the variables and observe the changes in the process using Simulator (Windschitl, M. And Andre, T., 1998). CPU-OS Simulator is free open source software which is a highly interactive, integrated and multi-level simulator developed specifically to support both the teachers and the learners of modern computer technologies at undergraduate level. This educational tool is designed for supporting the undergraduate lectures and the practical tutorial sessions in both the introductory and the advanced operating systems teaching modules (Besim Mustafa). This tool is useful for Compiler Construction, System Programming and Advanced Computer Architecture.

In this current study, we have designed an activity FBL (Free Open Source Software based Learning) for Operating System course where CPU-OS Simulator is used as free open source software.

## 2. Related Work

There are different ways to teach the course Operating System.

SOsim Simulator with visual facilities to serve as an effective tool for the better teaching and learning of concepts and techniques in modern operating systems was implemented (Maia, L.P. and Pacheco, A.C., 2003, 2005). The address translation simulator which was considered for the topic Virtual Memory in Operating System was used for both teaching and student evaluation (Robbins, S., 2005). An instructional operating system is used in teaching courses in operating systems (Charles L. A. and Nguyen, M., 2005). Naps, T.L., et. al. (2002) presented a framework for experimental studies of visualization effectiveness while Dann, W.(2001) used the visualization technique for programming course. A perspective based in constructivism to examine previous attempts at addressing problems associated with teaching operating systems was considered (David Jones and Andrew Newman). Instead of implementing operating systems components or modifying existing operating systems, a series of experiments that measure the performance of system services and try to infer information about their implementation from the results was conducted for the students (A.B. Downey, 1999). A possible way of using multithreaded programming to strengthen an operating systems course is presented (Ching-Kuang Shene, 2002). Think- Pair-Share (TPS) activity which consists of three phases Think-Pair-Share is a suitable strategy to use for instructors who intend to incorporate active learning techniques in their courses [Aditi Kothiyal et. al., Susan Ledlow, Carss and Wendy Diane, Sunita M Dol]. This activity is suitable to teach the Operating System (, 2016). The number of studies shows that technology tools like simulation has positive effect on learning outcome of students. (Jimoyiannis, A. And Komis, V,. 2001; Stern, L. et.al. ,2008 ; McKagan, S.B. et.al., 2009) In the current study, we considered the Free Open Source Software (FOSS) based Learning for Operating system and named the activity FBL (FOSS based Learning).

## 3. Methodology

FBL is conducted for Operating System of Third Year Computer Science and Engineering students. For this study, one group pre-post test method is considered.

A. Sample As Operating System is the course of third year Computer Science and Engineering, the group of 59 students is considered. This is one group pre – post test method.

B. FOSS (Free Open Source Software) Free and open-source software (FOSS) is computer software. It is classified as both free software and open-source software. Hence FOSS is freely licensed software to use, copy, study, and change the software in any way. The the source code of these softwares is openly shared so that people can improve the design of the software [https: //en.wikipedia.org/wiki/Free_and_open-source_ software].

C. Use of FOSS as a Visualization Tool Visualizations mean a variety of visual elements - Static Fig.s, Videos, Animations, Simulations, Interactive Animation/Simulation. FOSS is Simulator or animation tool which can be used in teaching-learning process.

D. CPU-OS Simulator CPU-OS Simulator is a free open source software that enables you to run programs manually created. It is possible to create a CPU program, enter CPU instructions in the program, run the program and observe and control simulations. It is also possible to access the part of the memory containing data so you can write or read data. The CPU-OS simulator is shown in the Fig. 1.

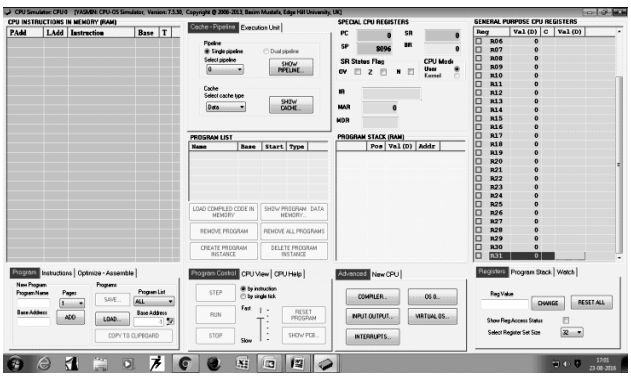

**Fig.1:CPU - OS Simulator**

E. What is FBL (FOSS Based Learning)?

In FBL, instructor explains the topic using FOSS CPU-OS Simulator. How to use the simulator CPU-OS for explaining the Scheduling algorithm- First Come First Served is illustrated in the following steps –

Step 1: Instructor launches the simulation tool and clicks on the compiler button shown in Fig. 2.

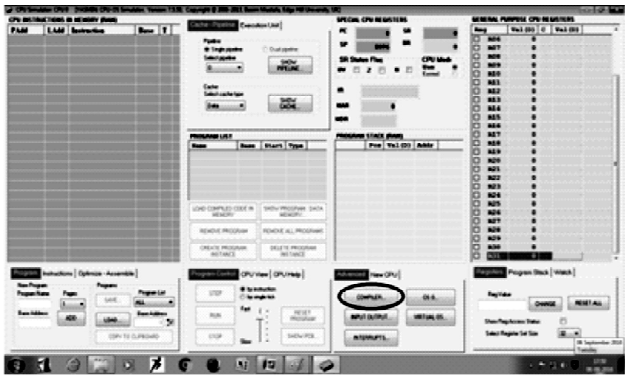

**Fig.2: CPU-OS Simulator main screen**

Step 2: Instructor writes the program in source section and compiles the written code to get the machine understandable program as shown in Fig. 3.

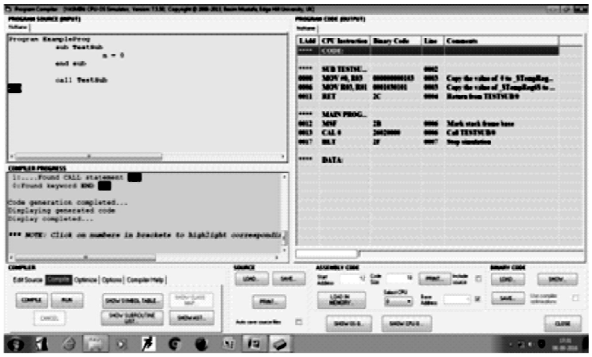

**Fig.3: Source program as well as compiled program in CPU-OS simulator**

Step 3: After step 2, Instructor loads machine language program into the main memory simply by clicking the button 'load program in the memory' as shown in Fig. 4.

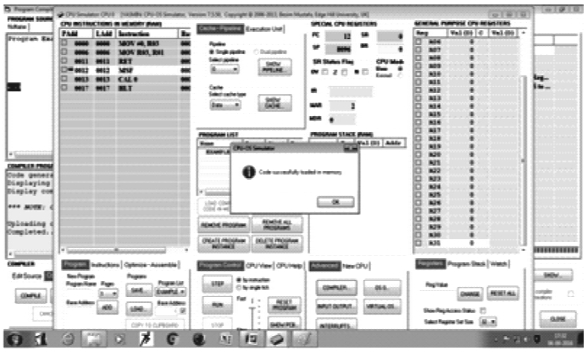

**Fig.4: Loading machine language program in main memory of CPU-OS Simulator**

Step 4: Instructor loads the OS (Operating System's) view to create several processes as well as to decide the scheduling policies. This particular task is shown in Fig. 5.

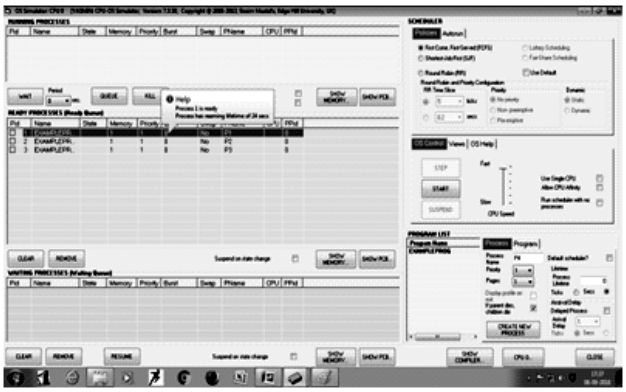

**Fig.5: Operating System's view to create processes on CPU-OS Simulator**

Step 5: After creating the several processes with specific parameters like process ID, priority, life time etc., we can have process states view shown in Fig. 6. This case explains the FCFS scheduling for uniprocessor system.

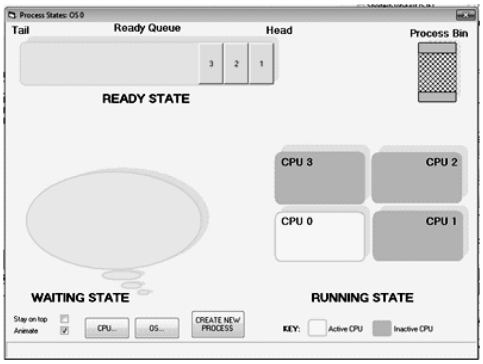


**Fig.6: Process states view on CPU-OS Simulator**

Step 6: Running snapshot of the previous case is shown in Fig. 7 where we have process P1 is in running state whereas P2 and P3 are waiting in the Ready queue for CPU (Central Processor Unit) to be get allocated.

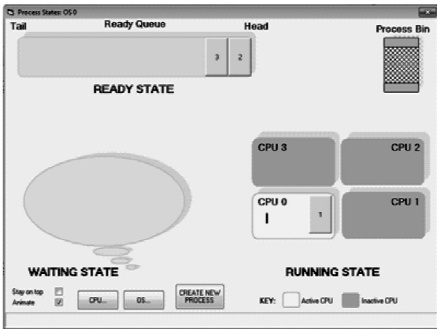


**Fig.7: Running snapshot of processes on CPU-OS Simulator**

Step 7: After successful execution and termination of all the created processes, we can have the results like waiting time, turnaround time etc. are shown in Fig. 8.

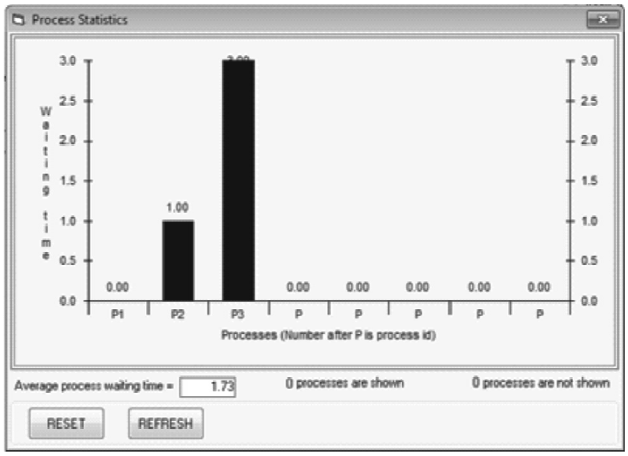


**Fig.8: Chart showing process related parameter on CPU-OS Simulator**

## 4. Experimental Detail

A. Experimental Set-up

The experimental set-up is given in Fig. 9.

| Techniques used | FBL - FOSS(Free Open Source Software): CPU-OS Simulator based Learning |
|---|---|
| Course | Operating System of Third Year |
| Sample | Third Year Computer Science and Engineering of Solapur University |
| Method | One group Pre and Post test model |
| Instruments used | 1. Pre-Test<br>2. Post-test<br>3. Survey questionnaire and feedback from open ended questions |
| Learning Domain used | 1. Bloom's Taxonomy<br>Cognitive Domain-Apply and Analysis Level |

**Fig.9: Experimental details**

B. Research Design

First the students were taught the scheduling algorithm topic which consists of four algorithms FIFO, SJF, Priority Scheduling and Round Robin algorithm using traditional method that is blackboard teaching. We conducted the pre-test which consists of three problems on these scheduling algorithms. The first problem was on the drawing the Gantt chart while two other on calculating the individual waiting time, average waiting time and average turnaround time for each scheduling algorithm. After this test, students were explained the scheduling algorithm using CPU-OS Simulator. The students were given post-test with similar types of the problems with different applications. The research design is shown in Fig. 10.

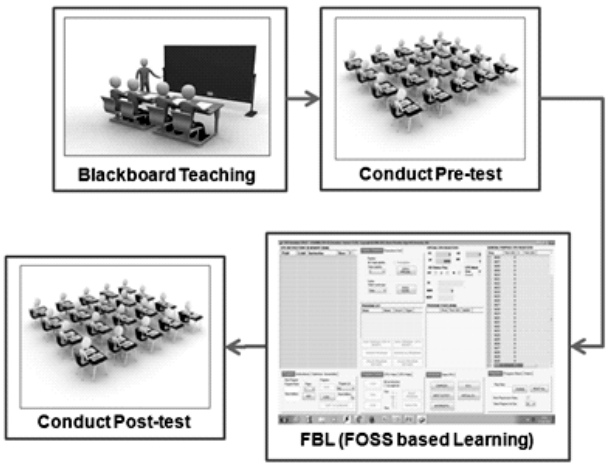


**Fig.10: Research design**

Learning Objectives (LO) of this study is to teach problem solving skill. These LOs are:

- To draw Gantt chart for four scheduling algorithms: First-in-First-out (FIFO), Shortest-Job-First (SJF), Priority Scheduling algorithm and Round Robin (RR) scheduling algorithm (LO1).
- To calculate waiting time of individual process and average waiting time of all processes for all four scheduling algorithms (LO2).
- To calculate average turnaround time for four scheduling algorithms (LO3).

Our research question (RQ) was

RQ: What is the difference between pre and post-test scores of the students for the problem solving after FBL?

Hypothesis of our study are

H1: Students' post test scores for LO1 are higher than pre-test score after teaching using FBL.

H2: Students' post test scores for LO2 are higher than pre-test score after teaching using FBL.

H3: Students' post test scores for LO3 are higher than pre-test score after teaching using FBL.

H4: Students' post test scores are higher than pre-test score after teaching using FBL

C-Pre and Post Test Questions

Bloom's taxonomy is a set of three hierarchical models used to classify educational learning

objectives into levels of complexity and mastery (https://en.wikipedia.org/wiki/Bloom%27s_taxonomy). Pre and Post Test questions covered Apply and Analyse Level of Cognitive domain of Bloom's Taxonomy. The question in test to satisfy all LOs is shown in the following Fig. 11.

**Consider three processes P1, P2, and P3. The CPU burst and priority values are given in the following table. A larger value means a lower priority.**

| | CPU Burst | Priority |
|---|---|---|
| P1 | 4 | 2 |
| P2 | 5 | 3 |
| P3 | 1 | 1 |

**a) Draw the Gantt chart for each scheduling algorithms FCFS, SJF, Priority and Round-Robin.**
**b) Calculate the waiting time for each process using four scheduling algorithm and average waiting time. Which of the schedules results in the minimal average waiting time?**
**c) Calculate the average turnaround time for each scheduling algorithm.Which of the schedules results in the minimal average turnaround time?**

---

ısider three processes P1, P2, and P3. The CPU burst a
ority values are given in the following table. A larger va
ıns a lower priority.

| | CPU Burst | Priority |
|---|---|---|
| P1 | 4 | 2 |
| P2 | 5 | 3 |
| P3 | 1 | 1 |

)raw the Gantt chart for each scheduling algorithms FCFS, S
ority and Round-Robin.
Calculate the waiting time for each process using f
eduling algorithm and average waiting time. Which of
edules results in the minimal average waiting time?
Calculate the average turnaround time for each schedul
orithm.Which of the schedules results in the minimal aver
around time?

| Table for Calculating Average Waiting Time | | | | |
|---|---|---|---|---|
| Scheduling Policy | Waiting Time | | | Average Waiting Time |
| | P1 | P2 | P3 | |

D. Feedback was conducted after FBL activity to understand the students' perception about the FBL activity. From the feedback given in Table 1, it is found that 92% students liked this activity.

**Table 1: Feedback about FBL**

| Sr. No | | Strongly Disagree | Disagree | Agree | Strongly Agree |
|---|---|---|---|---|---|
| 1 | I stayed interest in the content of the lecture because of the simulator | 7% | 20% | 68% | 5% |
| 2 | Discussing solution of the problem using CPU - OS Simulator helped me to learn the concepts | 2% | 15% | 70% | 13% |
| 3 | Example using CPU - OS Simulator helped in improving my knowledge. | 2% | 20% | 68% | 10% |
| 4 | Explaining the example with the help of simulator cleared the concept about the topic. | 3% | 20% | 60% | 17% |
| 5 | Simulator helped me in finding the solution to the problem. | 3% | 20% | 67% | 10% |
| 6 | Simulator is easy to understand and useful to make problem solving easier. | 2% | 13% | 73% | 12% |
| 7 | Simulator will be helpful at the time of examination. | 2% | 22% | 68% | 8% |
| 8 | I would not have learned as much from the lecture if there had been no simulator | 7% | 25% | 67% | 2% |
| 9 | Did you like the simulator based teaching? | **YES=92%** **NO=8%** | | | |

## 5. Result and Analysis

Students' understanding about the topic was analysed using pre-test, post-test marks for LO1, LO2, LO3 and overall marks. Test result is shown in Fig. 12 for LO1, Fig. 13 for LO2, Fig. 14 for LO3 and Fig. 15 for overall result. Graph in Fig. 12, 13, 14 and 15 shows that students performed better in post-test as compared to pre-test.

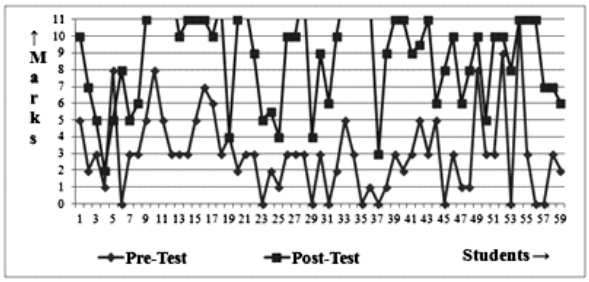


**Fig. 12: Pre-Post test comparison for LO1**

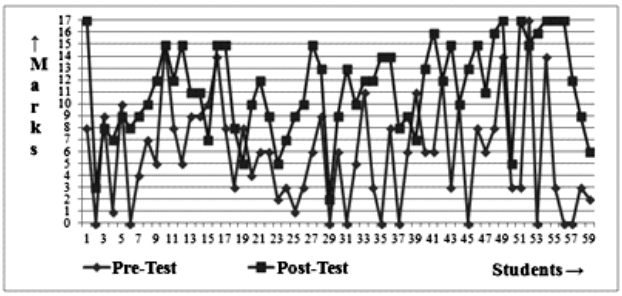


**Fig. 13: Pre-Post test comparison for LO2**

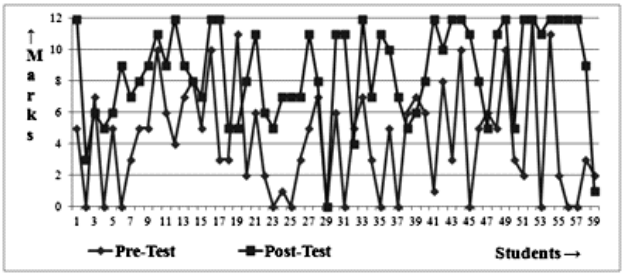


**Fig. 14: Pre-Post test comparison for LO3**

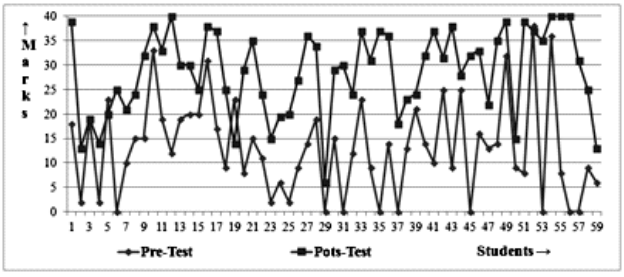


**Fig. 15: Pre-Post test comparison**

Statistical analysis is done using t-Test and Mann-Whitney test.

t-test is used to determine if two sets of data are significantly different from each other. For t-test to be significant statically, t must be at least 2.145 and p<=0.05.

Statistical analysis using t-Test is performed for pre-post test of LO1, LO2, LO3 and overall result.

t-Test result also shows statistical significant difference between pre and post-test result of students' performance for LO1, LO2, LO3 and overall result using FBL as shown in Table 2. Also the pre-post mean for LO1, LO2, LO3 and overall is given in Table 2.

**Table 2: Pre-Post Test means and t-Test Result**

| LOs | Pre- Test mean | Post- Test mean | t-Test Result | | | |
|---|---|---|---|---|---|---|
| | | | Degree of Freedom | Standard Deviation | t value | p value |
| LO1 | 3.1 | 8.8 | 116 | 2.67 | -11.7 | <0.0001 |
| LO2 | 5.8 | 11.3 | 116 | 4.14 | -7.22 | <0.0001 |
| LO3 | 4.2 | 8.6 | 116 | 3.24 | -7.30 | <0.0001 |
| Overall | 13.1 | 28.7 | 116 | 9.20 | -9.22 | <0.0001 |

We also performed the statistical analysis using Mann-Whitney test. It is a non-parametric test that is used to compare two population means that come from the same population. It is also used to test whether two population means are equal or not (http://www.statisticssolutions.com/mann-whitney-u-test/)

Mann-Whitney test using IBM's SPSS tool (https://www01.ibm.com/marketing/iwm/iwmdocs/tnd/data/web/en_US/trialprograms/W110742E06714B29.html) is shown in the Table 3. This test also shows the significant improvement in the performance of the students in the post-test compared to pre- test.

**Table 3: Statistical Analysis using Mann Whitney Test**

| LOs | Mann-Whitney Test |
|---|---|
| LO1 | Test Statistics [a]: Marks — Mann-Whitney U: 254.500; Wilcoxon W: 2024.500; Z: -8.045; Asymp. Sig. (2-tailed): .000 |
| LO2 | Test Statistics [a]: Marks — Mann-Whitney U: 618.000; Wilcoxon W: 2388.000; Z: -6.057; Asymp. Sig. (2-tailed): .000 |
| LO3 | Test Statistics [a]: Marks — Mann-Whitney U: 605.000; Wilcoxon W: 2375.000; Z: -6.144; Asymp. Sig. (2-tailed): .000 |
| Overall | Test Statistics [a]: Marks — Mann-Whitney U: 430.500; Wilcoxon W: 2200.500; Z: -7.056; Asymp. Sig. (2-tailed): .000 |

## 6. FBL for Other Courses

This FBL can be considered for other courses of Computer Science and Engineering. The courses and the FOSS available for other courses are given in the Table 4.

compared to post-test of LO1, LO2, LO3 and overall result. From the feedback it is also found that 92% students liked this activity. This activity can also be used for other courses given in Table 4.

**Table 4: FBL for other courses**

| Sr. No. | Course | FOSS available | Description |
|---|---|---|---|
| 1 | Theory of Computation | JFLAP | It is software for experimenting with formal languages topics including nondeterministic finite automata, nondeterministic pushdown automata, multi -tape Turing machines, several types of grammars, parsing, and L -systems, etc. [ Sunita M. Dol and Dr. S. A. Halkude , Susan H. Rodger , Vinay S. Shekhar ] |
| | | Visual Automata Simulator | A tool for simulating, visualizing and transforming finite state automata and Turing Machines. |
| 2 | Compiler Construction | JFLAP | It is software for experimenting with formal languages topics including nondeterministic finite automata, nondeterministic pushdown automata, multi -tape Turing machines, several types of grammars, parsing, and L -systems, etc .[ Sunita M. Dol and Dr. S. A. Halkude, Susan H. Rodger , Vinay S. Shekhar ] |
| | | Parsing emulator | Parsing simulator is a free open source software which implements the parsing table for a given grammar. This simulator generate the parsing table for LL1, SLR, LR and LALR.[ and Dattatray P. Gandhamal ] |
| 3 | Data Mining | Weka | The Weka workbench contains a collection of visualization tools and algorithms for data and predictive modeling, together with graphical user interfaces for easy access to this functionality. Weka supports several standard data mining tasks, data preprocessing, clustering, classification, association, visualization, etc.[ Sunita B Ah er and Lobo L.M.R.J.] |
| | | Tanagra | TANAGRA is free DATA MINING software for academic and research purposes. It proposes several data mining methods from exploratory data analysis, statistical learning, machine learning and databases area.[ http://eric.univ -lyon2.fr/~ricco/tanagra/en/tanagra.html ] |
| 4 | Java | Jeliot | Jeliot 3 is a Program Visualization application. It visualizes how a Java program is interpreted. Method calls, variables, operation are displayed on a screen as the animation goes on, allowing the student to follow step by step the execution of a program[ http://cs.joensuu.fi/jeliot/ ]. |
| 5 | Object Oriented Programming | Alice | Alice is a freely available teaching tool designed to be a student's first exposure to object -oriented programming. It allows students to learn fundamental programming concepts in the context of creating animated movies and simple video games.[ http://www.alice.org/index.php ] |
| 6 | Operating System | CPU-OS Simulator | CPU-OS simulator is a free program that ena bles to run program manually created. It is possible to create a CPU program, enter CPU instructions in the program, run the program and observe and control simulations with the help of this tool[ http://www.teach -sim.com/ ] |
| 7 | Advanced Computer Architecture | | |
| 8 | System Programming | | |
| 9 | Computer Organization | | |
| 10 | Network Security | Snort | Snort is a free and open source network intrusion prevention system (NIPS) and network intrusion detection system (NIDS) [ https://www.snort.org/ ]. |
| | | Wire shark | Wireshark is a free and open -source packet analyzer which is used for network troubleshooting, analysis, software and communications development, and education . [ https://www.wireshark.org ]. |

## 7. Conclusions

In the present study, we have designed the FBL i.e. FOSS (Free Open Source Software) based Learning for the course Operating System of Third Year Computer Science and Engineering. The CPU – OS simulator is used as FOSS in this activity. The statistical analysis using t-Test and Mann-Whitney test showed the significant improvement in pre-test as

## References

Van Berkum et. al. (1991) Instructional Environment for simulations. Educations and Computing, 305,358.

Windschitl, M. And Andre, T. (1998) Using computer simulation to enhance conceptual change: the role of constructivist instruction and students epistemological beliefs.

A.B. Downey (1999) Teaching experimental design in

an operating systems class, Proc. 30 ACM SIGCSE.

Susan Ledlow (2001) Using Think-Pair-Share in the College Classroom

Jimoyiannis, A. And Komis, V,. (2001) on in Physics teaching and learning: a case study ob students' understanding of trajectory motion. Computers and Education, 36(2), 183,204

Dann, W., Cooper, S., and Pausch, R. Using Visualization To Teach Novices Recursion. 6 th Annual ACM SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education (ITiCSE 2001), Canterbury, England (June 2001), 109–112

Naps, T.L., et. al. 2002. Exploring the Role of Visualization and Engagement in Computer Science Education. ACM SIGCSE Bulletin 35 (2), June 2003.

Ching-Kuang Shene (2002) Multithreaded Programming Can Strengthen an Operating Systems Course, accessed from https://www.cs.mtu.edu/~shene/PUBLICATIONS/2002/os.pdf on 25-08-2016

Maia, L.P. and Pacheco, A.C. (2003) A Simulator Supporting Lectures on Operating Systems, 33rd ASEE/IEEE Frontiers in education Conference, November 5-8, 2003, Boulder, CO.

Charles L. A., Nguyen, M. (2005) A Survey of Contemporary Instructional Operating Systems for Use in Undergraduate Courses, JCSC 21 (1), pp. 183-190.

Luiz Paulo Maia et. Al.(2005) A Constructivist Framework for Operating Systems Education: a Pedagogic Proposal Using the SOsim, ITiCSE'05, June 27–29, 2005, Monte de Caparica, Portugal. Copyright 2005 ACM 1-59593-024-8/05/0006

Robbins, S. (2005) An address translation simulator. SIGCSE'05 February 23-27, 2005, St. Louis, Missouri, USA.

Carss and Wendy Diane (2007) The Effects of using Think-Pair-Share during Guided Reading Lessons, The University of Waikato.

Stern, L. et.al. (2008) The effect of the computerised simulation on middle school students' understanding of the kinetic molecular theory, Journal of science Education and Technology, 17(4),305-315.

McKagan, S.B. et.al. (2009) A research based curriculum for teaching the photoelectric effect, American Journal of Physics, 77(1), 87-94

Silberschatz, A., Galvin, P.B., Gagne, G. (2010) Operating System Concepts. 8th Edition. Wiley.

Besim Mustafa (2011) Visualizing the Modern Operating System: Simulation Experiments Supporting Enhanced Learning, accessed from http://repository.edgehill.ac.uk/4735/1/Paper_Final_Version_ACM_ITE_New_York_2011.pdf on 23-08-2016

Sunita B Aher and Lobo L.M.R.J. (2011) Data Mining in Educational System using WEKA, published in International Conference on Emerging Technology Trends (ICETT), Kollam

Rutten, N. et. al. (2012) The learning effect of the computer simulation in science education. Computer and Education, 58(1), 136-153

Besim Mustafa (2013) YASS: A system simulator for operating system and computer architecture teaching and learning, European Journal of Science and Mathematics Education Vol. 1, No. 1

Aditi Kothiyal et. al. (2013) Effect of Think-Pair-Share in a Large CS1 Class: 83% Sustained Engagement, ICER'13, http://www.it.iitb.ac.in/~sri/papers/tps-icer2013.pdf.

Sunita M. Dol and Dattatray P. Gandhamal (2014) TPFOSSS: A Modified TPS Technique to Improve Student's Conceptual understanding of Compiler Construction Course, in 6th IEEE International Conference on Technology for Education (T4E-2014), Amrita University, Kerala, India.

Dr. S. A. Halkude and Sunita M. Dol (2015) An Active Learning Strategy Think-PairFree Open Source Software-Share to Teach Engineering Courses", in 2nd Inetrnational Conference on Transformation on Engineering Education (ICTIEE-2015), Bangalore, India. Paper is in Journal of Engineering Education Transformation, ISSN 2349-2473 (Print), eISSN 2394-1707 (Online).

Komal R. Pardeshi (2016) Improving the Student Performance Using Think-Pair-Share for Operating System, Journal of Engineering Education and Transformation, DOI: 10.16920/jeet/2016/v0i0/85651

David Jones and Andrew Newman, A Constructivist-based Tool for Operating Systems Education, accessed from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.87.7151&rep=rep1&type=pdf on 25-08-2015

http://eric.univlyon2.fr/~ricco/tanagra/en/tanagra.html

http://rapid-i.com

www.jflap.org

Susan H. Rodger, Increasing the Use of JFLAP in

Courses accessed from

https://users.cs.duke.edu/~rodger/talks/pvw2011/talkpvw2011jflap.pdf on 25-08-2016

Vinay S. Shekhar et. al. , Enhancing JFLAP with automata construction problems and automated feedback, accessed from

https://www.researchgate.net/publication/282738976_Enhancing_JFLAP_with_automata_construction_problems_and_automated_feedback on 25-08-2016

http://cs.joensuu.fi/jeliot/

http://www.alice.org/index.php

http://www.teach-sim.com/

https://www.snort.org/

https://www.wireshark.org

https://en.wikipedia.org/wiki/Free_and_open-source_software

http://www.teach-sim.com/

https://en.wikipedia.org/wiki/Bloom%27s_taxonomy

https://en.opensuse.org/Free_and_Open_Source_Software

http://www.statisticssolutions.com/mann-whitney-u-test/

https://www.01.ibm.com/marketing/iwm/iwmdocs/tnd/data/web/en_US/trialprograms/W110742E06714B29.html