DeCAIR Course Syllabus Form

Author(s)	Ramzi Saifan		
Author Organization Name(s)	The University of Jordan		
Work Package Number & Title	Work Package 6: Development of existing BSc programs in AIR		
Activity Number & Title	Activity 6.1: Designing and developing syllabi and content for the agreed upon courses in the new programs		
Work Package Leader	Jorge Casillas, University of Granada		
Due Date of Delivery	1/2/2022	**Project Month**	M14
Submission Date	11/4/2021	**Project Month**	M11

Revision History

Version	Date	Author	Description	Action *	Page(s)
1	11/4/2021	Ramzi Saifan	Original (base) document	C	1-5
2	4/12/2022	Samah Rahamneh	Update	U	1-5
3					
4					

(*) Action: C = Creation, I = Insert, U = Update, R = Replace, D = Delete

Disclaimer

This project has been co-funded by the Erasmus+ Programme of the European Union.

Email: DeCAIR@ju.edu.jo

Project Website: http://DeCAIR.ju.edu.jo/

The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Course title	Modern Operating Systems
Course number	0907443
Credit hours (lecture and lab)	3 (3 + 0)
ECTS (weekly contact and self-study load)	6 (3 + 3)
Prerequisites/co-requisites by course number and name	0907346 Data Structures and Algorithms
Prerequisites by topic (other than the formal prerequisites above)	Students are assumed to have had sufficient knowledge pertaining to algorithms and one programming language.
Level and type (compulsory, elective)	Bachelor compulsory course
Year of study and semester	Third year, second semester
Catalogue description	The goal of this course is to provide an introduction to the internal operation and structure of modern operating systems. In particular, Theories and implementation of modern operating systems including operating system interface (system calls), process and thread management, CPU and disk scheduling, synchronization, deadlock, memory management and virtual memory, file system, device management and I/O handling. Case studies for modern operating systems such as Android and iOS. Overview of Robotic Operating Systems (ROS). Students also gain hands-on experience on both the user and developer sides of open-source operating systems such as Linux.
Objectives	1. Recognize the importance of the operating systems. 2. Recognize the interaction between the applications and the operating system. 3. Understand different resources management such as: processors, memory and I/O. 4. Understand different scheduling algorithms used by operating systems. 5. Recognize different types of operating systems: desktop, mobile, embedded and robotic operating systems.
Intended learning outcomes	Upon successful completion of this course, students will be able to:

No	Intended learning Outcome (ILO)	Program learning outcome (PLO)*
1	Demonstrate knowledge and understanding of the different modules in a modern Operating System (OS) in general and memory management.	7
2	Exemplify and explain how the kernel of an OS is designed, including being able to explain what a	7

<table>
<tr><td></td><td>
<table>
<tr><td></td><td>process is, the interaction between the kernel and the hardware, user mode vs. kernel mode and process management.</td><td></td></tr>
<tr><td>3</td><td>Design and construct the following OS components: System calls, Schedulers, Memory management systems, Virtual Memory and Paging systems.</td><td>2</td></tr>
<tr><td>4</td><td>Demonstrate knowledge and understanding of the design of different types of modern operating systems including desktop, mobile, embedded and robotic operating systems.</td><td>7</td></tr>
</table>
(*) The PLOs are listed in the appendix
</td></tr>
<tr><td><b>Teaching and learning methods</b></td><td>Development of ILOs is promoted through the following teaching and learning methods:
<ul>
<li>Lectures will be in class.</li>
<li>The student attends the class presentations and participates in the discussions.</li>
<li>The student joins the related online team/group and participates in its discussions.</li>
<li>The student studies the reference material, including books and videos.</li>
<li>The student solves the programming assignments that include modifying the Linux kernel and implementing scheduling algorithms.</li>
</ul>
</td></tr>
<tr><td><b>Learning material type</b></td><td>Textbook, class handouts, some instructor keynotes, selected YouTube videos, and access to a personal computer and the internet.</td></tr>
<tr><td><b>Resources and references</b></td><td>A- Required book(s), assigned reading and audio-visuals:
<ol>
<li>Silberschatz, Galvin, and Gagne. Operating System Concepts. John Wiley & sons , inc. 10th Edition (April 2018)</li>
</ol>
B- Recommended book(s), material and media:
<ol start="2">
<li>Modern Operating Systems by Andrew S. Tanenbaum, Pearson; 4th edition (March 10, 2014)</li>
<li>Operating systems design and implementation, Andrew s. Tanenbaum, Pearson; 3rd edition (January 4, 2006)</li>
<li>Robot Operating System (ROS) by Anis Koubaa, Springer; 1st ed (July 18, 2021)</li>
</ol>
</td></tr>
<tr><td><b>Topic outline and schedule</b></td><td>
<table>
<tr><th>Week</th><th>Topic</th><th>ILO</th><th>Resources</th></tr>
<tr><td>1</td><td>Introduction / Operating-System Structures</td><td>1+2</td><td>1</td></tr>
<tr><td>2</td><td>Processes</td><td>1+2</td><td>1</td></tr>
<tr><td>3+4</td><td>Multithreaded Programming</td><td>1+2+3</td><td>1</td></tr>
<tr><td>5+6</td><td>Process Scheduling</td><td>1+2</td><td>1</td></tr>
<tr><td>7</td><td>Synchronization / Deadlocks</td><td>1+2</td><td>1</td></tr>
</table>
</td></tr>
</table>

8+9	Memory Management	1+2	1
10+11	Virtual Memory	1+2	1
12	File Systems	1+2	1
13	I/O Systems	1+2	1
14	Robotic Operating Systems (ROS)	7	4

Evaluation tools

Opportunities to demonstrate achievement of the ILOs are provided through the following assessment tools:

Assessment tool	Mark	Topic(s)	Time
Homework assignments	15%	Programming aspects	W2-W14
Quiz	5%	Modern operating systems structure and Processes	W5
Midterm exam	30%	Modern operating systems structure, Processes, threads, and CPU scheduling	W8
Final exam	50%	All material	W16
Total	**100%**		

Student requirements

The student should have a computer and internet connection.

Course policies

A- Attendance policies:

- Attendance is required. Class attendance will be taken every class and the university polices will be enforced in this regard.

B- Absences from exams and not submitting assignments on time:

- A makeup exam can be arranged for students with acceptable absence causes.
- Assignments submitted late, but before announcing or discussing the solution can be accepted with 25% penalty.
- The project report must be handed in in time.

C- Health and safety procedures:

- All health and safety procedures of the university and the school should be followed.

D- Honesty policy regarding cheating, plagiarism, misbehavior:

- Open-book exams
- All submitted work must be of the submitting student.
- Other text or code must be properly quoted with clear source specification.
- Cheating will not be tolerated.

E- Available university services that support achievement in the course:

- Microsoft Teams team and Moodle course page

	• CPE Labs for practicing the practical aspects and solving the programming assignments. • Program announcements Facebook page
Additional information	None

Appendix

Learning Outcomes for the BSc in Computer Engineering

Students who successfully complete the BSc in Computer Engineering will be able to:

[1]	an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics
[2]	an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors
[3]	an ability to communicate effectively with a range of audiences
[4]	an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts
[5]	an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives
[6]	an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions
[7]	an ability to acquire and apply new knowledge as needed, using appropriate learning strategies