EST 1892 LSBU

Coursework Specification

Read this coursework specification carefully, it tells you how you are going to be assessed, how to submit your coursework on-time and how (and when) you'll receive your marks and feedback.

Module Code	CSI_5_OSY
Module Title	Operating Systems
Lecturer	Ioannis Iatropoulos
% of Module Mark	60%
Distributed	28/02/2024
Submission Method	Submit online via this Module's Moodle site
Submission Deadline	22/03/2024 and 23:30
Release of Feedback & Marks	Feedback and provisional marks will be available in the Gradebook and Turnitin (Rubric) on Moodle within 2-3 weeks from the submission deadline.

Coursework Aim:

This coursework requires the students to undertake an individual project, which requires them to install and use MS-DOS emulator implemented on a Debian OS, by using VirtualBox as a Hypervisor.

On this assessment, students are required to obtain Debian-OS from the Official Website (see resources) and install it on a Virtual Machine with their preferred virtualized hardware. Within this virtual machine an MS-DOS emulator must be installed, and students must execute multiple games and applications, and measure its performance. This evaluation must include benchmarking and resource monitoring, evaluate key performance indicator markings, and examine any system scalability. Then submit a scientific report individually, along with any supporting materials, and demonstrating their project in a video presentation.

This coursework will test the students on their knowledge and understanding of MS-DOS operating system, managing and monitoring resources, handling processes and threads, allowing them to demonstrate these skills in a practical test case by utilizing a Hypervisor.

(See Assignment Task, below).

Coursework Details:

Type:	Project Report, Group Presentation, Project materials
Word Count:	A minimum of 2500 words is recommended, with 5000 words maximum limit. The following texts will be excluded from the word counts. • Footnotes for reference purposes only. • Bibliography. • List of tables, Figures and Table of Contents. • Source code and scripts. • Everything that is included in the Appendix section. You might append source codes, development environments, and any additional resources at the appendix section. However, these will be excluded from the marking scope. Please note in case you develop you own tool or code; the complete source code must be submitted as a separate script for evaluation and correctness as supporting materials.
Document Structure and Readability:	▪ The document is **Recommended** to follow the format from one of the suggested *Project report templates* provided under the *Learning and Support* section on VLE (https://overleaf.com). ▪ Work must be cited through appropriate bibliography. ▪ Work must be submitted as a Word document (.doc/docx) or a PDF (not exceeding 100MB). ▪ The coursework document must follow a consistent template, using a single font and font size (e.g., Arial 11 or larger if you need to for the headings), with a recommended 1.5 line spacing. ▪ Any code analysis or command execution must be complete and concise with proper commenting to explain the logic, attributes/options/flags and input parameters. To present the **Source Code** snippets, command execution, bash scripts, or configuration files, you need to use a fixed-width font and clearly readable (no screenshots), recommended Monospaced fonts (Consolas, Courier New, etc.). You might split the code by module and present it through a table (1x1) used as a code-box, that will clearly separate from the narrative. ▪ Your student ID number must appear on the front page of the coursework. Your name must not be on your coursework.

Screenshots Requirements:	▪ Each student after installing the Debian Operating System in the VirtualBox, must create a new system user with their username along with their **last two digits** of their student-ID. The username could be your first name, your surname or a combination from the first letters of both, for example “ioaniatr80”. (Example at the end of this document) ▪ The screenshots need to include your user on each command execution, unless the command requires the ‘root / administrator’ user and cannot be executed with “sudo / run as administrator”. (Example at the end of this document) ▪ **Screenshots** must have a captioned, cross-referenced and should be used mostly to provide evidence of: o Output or results of a program/tool execution (any command executed as an input must be provided also by text). o Designs, such as Information on the network environment, IP addresses, activity diagrams/flowcharts etc. o Present a software interface (GUI/CLI), or tools. o Graphical Analysis may include a representation of the results or similar types of comparative plots (e.g., for resource monitoring or performance evaluation). o Any other relevant evidence that can be used as proof of concept. Note: Screenshots which do not satisfy the above-mentioned requirements will not be considered and the report will be penalized accordingly!
Referencing:	Harvard Referencing should be used, see your Library Subject Guide for guides and tips on referencing. The references must include **at least two** (or more) of following types of sources: (Mandatory) A reference with the link of the selected project and its origin, along with the name of the application/software (if applicable). (Mandatory/Optional) References of any installation tutorials or usage guides that has been followed. (Optional) References of any forums or links followed,

	along with access date and title. (Optional) Books and other resources studied to accomplish these tasks. Note: Inappropriate reference formats or mixed referencing will be penalized as stated in the assessment rubric's report structure and readability criteria!
The Content:	• The final report must document all the steps, commands issued, and console output in the form of a scientific format. • The documentation should be thorough enough that the implementation can be replicated step-by-step by a technically incompetent reader. Note: Your report should be narrative in style, with human explanation and commentary. A "report" that is merely a collection of screenshots and data dumps will be graded very poorly.
Video Presentation:	The video presentation must include a summary of your work in the form of a short video (.mp4), duration up to 10 minutes max, explaining your work, and a sample demo is required.
Submission:	Students could submit up to 3 different FILES: i. (Mandatory) The main report (.doc, .docx, .pdf), which includes the documentation of the coursework in a scientific format. ii. (Optional) A compressed file (.zip), with the supporting materials e.g., source code or scripts that has been developed (.sh, .py, .cpp, .java), tool or release versions (executable file .pyc, .exe, .out, .jar etc.) that are not included into the default Kali-Linux installation, references/links with code snippets that has been used, dependencies (e.g., libraries/directives), or other kind of software that is required to recreate the testing environment etc. In case the file size exceeds 100MB, cloud storage should be used (OneDrive, Google Drive, etc.) with appropriate access rights, and submit a single text file (.txt) that will include the link. This is optional, as it is supposed that those materials should be included in the SD-Card that needs to be delivered to your tutor at the end of the project. Supporting materials are required to satisfy some of the criteria as implementation evidence.

	iii. (Mandatory) The audiovisual presentation (.mp4), summarizing the work. Alternatively, in the case of a large video file, share it via a cloud solution (e.g., OneDrive, YouTube etc.). In this case, the link should be submitted through a simple text file (e.g., “presentation.txt”). Note: Any submission with inaccessible material(s) will not be considered for marking. The filenames need to have the format: i. studentID_name-coursework2_report.doc (example: 123456_John_Doe-coursework2_report.doc) ii. studentID_name-project_materials.zip (example: 123456_John_Doe-project_materials.zip) iii. studentID_name-presentation.mp4 (example: 123456_John_Doe-presentation.mp4)
Regulations:	Make sure you understand the University Regulations on expected academic practice and **academic misconduct**. **Note in particular:** ▪ Your work must be your own. Markers will be attentive to the plausibility of the sources provided and the consistency and approach to writing the work. Simply, if you do the research and reading, and then write it up on your own, giving reference to sources, you will appropriately approach the work appropriate way and will not give markers reason to question the authenticity of the work. ▪ All quotations must be credited and properly referenced. Paraphrasing is still regarded as plagiarism if you fail to acknowledge the source for the ideas being expressed. **TURNITIN:** When you upload your work to the Moodle site, it will be checked by anti-plagiarism software. Your similarity index for the report must not be more than 20%. Any report with exceeding a 20% similarity index will be subject to Academic Misconduct Investigation (AMI).

Learning Outcomes

This coursework will fully or partially assess the following learning outcomes for this module.

- Understanding of Hardware and Software Virtualization.
- Learning about software emulators and applications tailored for them.
- Understanding of Operating Systems principles, including how processes and resources can be managed for various applications.
- Developing problem-solving ability through experimentation and testing, with appropriate applications and ROMs setup.
- Developing troubleshooting skills to Identify and fix issues that may arise during the project implementation.
- Acquiring project planning and management skills, including setting goals, defining requirements, and scheduling.
- Have a solid understanding of operating system designs and strategies for a large range of monitoring functionalities.
- Synthesize and utilize resources/tools to efficiently and solve the given tasks.
- Learning to document the project, including design, implementation, and results.
- Demonstrate a systematic understanding of Operating Systems for virtualized devices based on practical experience.

Assessment Tasks

This coursework is individual based and weighs **60%** of the overall marks.

The primary objective of this assessment is to run MS-DOS operating system by using an emulator within a Debian OS running on a virtual machine. Perform several tasks as monitor resources, execute applications and games, benchmarking and performance evaluation, aiming to understand how the operating system is managing those resources. It is also expected to demonstrate their application in a short video duration 10 minutes maximum.

The report must be submitted individually, written based in their own experience, and demonstrating understanding of what has been achieved. The report must clearly present student strengths or weaknesses. All reports will be subjected to similarity analysis to identify students who have not written in their own words but used the same writing as others.

Video Presentation: The video will be evaluated based on the following criteria:

- Presentation format / consistency
- Duration
- Coverage and scope
- Fluency and clarity
- Domain of knowledge
- Demonstration / Soft simulation

The coursework assessment is mainly divided into two tasks as follows.

Task 1 – Application Implementation

For this task, students require to accomplish the following:

- **Domain of Knowledge:** Students must show an in-depth understanding of the MS-DOS operating system and its applications by describing its features. Moreover, must include a discussion of how the program relates to the theoretical concepts covered in the module, by presenting this application from the aspect of the operating system. Explain how the MS-DOS operating system handles the applications, interacting with hardware/software resources, processes, and threads that are spawned (if any), users involved, access level (user/kernel), locations in the file system used, examine any concurrency issues, present requirements, design system model and application workflow etc.
- **MS-DOS Emulator:** Students are required to install an MS-DOS emulator (such as DOSBox) on the Debian Linux distribution. The emulator must NOT be installed on any other derivative Linux distribution, nor on the host machine. Debian-OS must be installed on VirtualBox as the Hypervisor, with students selecting their preferred settings for virtualized resources based on the available hardware. Furthermore, students must install and configure the system effectively, utilizing the operating system's facilities, and importing Game-ROMs or applications developed for MS-DOS. There are several places to get DOS games online, with hundreds of free classic games/ROMs that were abandoned by their developers. (see resources).

Task 2 – Evaluation of the MS-Dos Emulator

The Debian-OS from the previous task must be examined thoroughly based on use of resources, and evaluate the MS-DOS emulator performance and scalability, even under different circumstances. More specifically:

- **Resource Monitoring:** On the Debian-OS, students must perform resource monitoring and benchmarking over a period of time, identify the key performance indicator markings (CPU Utilization, GPU Utilization etc.) for the MS-DOS emulator, and create an analysis and visualizations of resource monitoring while executing several games/ROMs. Students may need to synthesize various resources or monitoring tools to accomplish this task.
- **Performance evaluation:** On the MS-DOS emulator, evaluate the effectiveness of the MS-DOS based on its role and nature. Provide means to measure the performance of the system, evaluate and assess the system in use by conducting a benchmark tests. For this purpose, various metrics must be tested on cases/profiles that will be created, such as, under heavy workload (of the Debian or Host-OS), utilizing multiple Debian-VMs, Multiple Emulators, multiplayer games etc. Finally, students must interpret the results and provide a critically evaluative analysis discussing their findings or any scalability of the system. It is recommended to be used resource intensive MS-DOS games such as “Redneck Rampage”, “Need for Speed” etc. along with DOS Benchmark tools (see resources).

Assessment Criteria and Weighting

The report will be marked using the marking criteria provided in this coursework specification document. All the students are advised to fully understand the marking criteria before starting the coursework.

LSBU marking criteria have been developed to help tutors give you clear and helpful feedback on your work. They will be applied to your work to help you understand what you have accomplished, how any mark given was arrived at, and how you can improve your work in future.

Marking Criteria

	Criteria		Feedforward comments				
		Weighting	Excellent (100-71%)	Good (70-61%)	Pass (60-51%)	Weak (50-41%)	Poor (40-0%)
	Report and Documentation						
	Report Structure, Readability, and Referencing **(Marks 5%)**	5%	Sophisticated, consistent, error free application of relevant topics conventions with great attention to detail. Excellent writing, structure, spelling, grammar. The length of the report is more than 2500 words. The references are complete and in the correct format. Used correctly inside the text. Provide valid and reliable information.	Comprehensive application of relevant topics conventions with few errors. Very good writing, structure, spelling, grammar, but with minor errors. The length of the report is more than 2500 words The references are valid and complete, but there are some errors in the format.	Generally correct application of relevant topics conventions, with some errors and / or inconsistencies. The length of the report is at least 2500 words. Sufficiently written with little structure, spelling, grammar with some errors. All sources are valid and reliable, but the references are incomplete or partially used inside the text.	Poorly written with confusing structure, spelling, grammar and / or errors. Below the minimum of 2500 words. The references are incomplete, or no references used inside the text. They include some sources that are not valid or reliable.	Poorly written, less than 2500 words, with no academic style, structure, spelling, grammar and/or multiple errors. The final report does not include any references. Most of the references are not valid.
	Abstract and Introduction **(Marks 5%)**	5%	A well-articulated abstract and introduction that provides a clear, logical, and succinct description of content, objectives, scope and requirements. The organization of the review, which draws the reader's attention to a central concern, debate, or contention.	A well-articulated abstract and introduction that provides a clear, logical description of content, objectives, scope, requirements and organization of the review	Satisfactory abstract and introduction that has a good reflection and description of the content, objective, scope, and organization of the report.	An abstract that articulates some key components of the report. An introduction that outlines the content, scope, and organization of the report	Either no abstract or introduction, or one that poorly or partially situates the reader in the context of the concern, debate, or contention addressed in the report
	Conclusion and Recommendations **(Marks 5%)**	5%	Excellent breadth, accuracy, and detail in understanding key aspects of subject and reflection of the project. Contributes to subject debate. Very good awareness of ambiguities and limitations of knowledge. Provides high-level summary and recommendations are very accurate and detailed.	Good depth of understanding of key aspects of subject, and reflection of the project is shown Evidence of coherent knowledge. Very good contribution to subject debate. Very good summary and accurate recommendations	Demonstrated good understanding of key aspects of subject, with a good reflection of the project. Some evidence of coherent knowledge and own critique. Sufficiently summarize the report with some recommendations	Weak evidence of superficial understanding of subject, with Inaccuracies. Some outline of recommendations, does not summarize well the report or lack of succinctness. Not good reflection of the project.	Little or no evidence of understanding of subject. Inaccuracies. Without any recommendation, does not summarize the report.
	Application Implementation						
	Domain of Knowledge **(Marks 10%)**	10%	Shows breadth, accuracy and detail in understanding key aspects of subject. Contributes to subject debate. Some awareness of ambiguities and limitations of knowledge. Knowledge and understanding are consistent and accurately developed with a level of criticality. Exceptional error-free presentation of the software/hardware, and system requirements, with extensive appropriate use of notation. System model and design diagrams provide a comprehensive and insightful set of views of the systems use.	Accurate and extensive understanding of key aspects of subject. Evidence of coherent knowledge. Knowledge and understanding are basic/relatively superficial. A complete and appropriate use of notation to present software/hardware, and system requirements. The System model and design diagrams accurately represent almost all the usage requirements of the system in use.	Accurate understanding of key aspects of subject. Evidence of coherent knowledge. Knowledge and understanding are detailed and satisfactory. Simple design diagrams with reasonable use of notation with some errors. The system model accurately represents the basic use of the system.	Some evidence of superficial understanding of subject. Minor Inaccuracies. Knowledge and understanding shows consistent gaps. Limited use of notation with obvious errors and mistakes. The system model and design diagrams provide a poor or misleading interpretation of the use of the system.	Little or no evidence of understanding of subject. Many Inaccuracies. Knowledge and understanding are poor and lacks academic rigour. Hardly any information provided about the system model.

	Implementation **(Marks 15%)**	15%	Excellent problem-solving ability and implementation of the proposed methodologies and solutions. Ability to Adapt to unforeseen practical and theoretical challenges to achieve project objectives. Well-crafted technical solution, addressing all aspects of the user requirements.	Very good problem-solving ability and implementation of the proposed methodologies and Solutions. Adapt to practical and theoretical challenges to achieve project objectives. Comprehensive technical solution, addressing various aspects of the user requirements.	Sufficient problem-solving ability and implementation of the proposed methodologies and solutions. Some adaptation to practical and theoretical challenges to achieve project objectives identified goals. Good technical solution, addressing most aspects of the user requirements.	Limited problem-solving ability and implementation of the proposed methodologies and Solutions. Limited exploration of possible solution(s) using established approaches to resolve practical and theoretical problems. Weak attempt at the technical solution, addressing only few aspects of the user requirements.	Poor or lack of problem-solving ability and implementation of the proposed methodologies and solutions. Little or no exploration of solution(s). Question or problem unresolved. Poor attempt at technical proposition.
	Resource Monitoring **(Marks 20%)**	20%	Very high-quality analysis of the system developed independently. Sustained benchmarking, resource monitoring and synthesis of tools/practices. Use of evidence-based arguments. Thoroughly identifies and explains the key indicator markings. States the system status with extensive level of detail for the handling of the related processes, and other components.	Very good system analysis and use of tools/practices for benchmarking and monitoring of resources. Use of evidence-based arguments. Well identifies most of the key indicator markings, and states the system status on handling processes, data or other components.	Applies practical and/or technical skills correctly and with some application for resource monitoring. Adapts to unforeseen practical and theoretical challenges to achieve identified possible key indicator markings, and sufficiently present the system status.	Some evidence of limited problem-solving ability. Appropriate response to monitor resources, but there is little development and exploration evident of the possible solution(s), and/or weak interpretation of the key indicator markings. Some attempt at evaluation and synthesis of resources. Some use of evidence-based arguments.	Little or no exploration of possible solution(s), no exploration of the available resources. Very little use of evidence-based arguments to present the status of the system.
	Performance Evaluation **(Marks 20%)**	20%	An exceptional level of embodied awareness revealed in a highly sensitive and detailed performance evaluation. Examines all the aspects for scalability of the system.	Well-developed embodied awareness; for the most part. A detailed and accurate responsiveness; Very good use of tools/practices for a dynamic performance evaluation and interpretation of the results. Examines most of the aspects for scalability of the system.	Applies practical and/or technical skills correctly and with a proper use of tools and applications for performance evaluation, that may include minor inaccuracies in the interpretation of the results. Satisfactory presentation of the system's scalability.	Some evidence of limited problem-solving ability. Appropriate response to evaluate performance, but there is little development and exploration evident of the possible solution(s). Weak interpretation of the results, performance evaluation, or scalability of the system.	Little or no exploration of possible solution(s) for performance evaluation, or scalability of the system.
	Presentation						
	Video Presentation **(Marks 20%)**	20%	Showed excellent confidence & composure. Very clear, persuasive and compelling with skilful use of the presentation format. Presentation addresses the needs of the audience very well. Excellent demonstration of the system, with consistence understanding for the domain of knowledge.	Were mostly confident & composed. presentation is clear, mostly persuasive, compelling and skilfully presented. Presentation addresses demonstration and the needs of the audience to a large degree.	Good use of the presentation format and skills. Presentation considers the needs of the audience, with basic demonstration of the system.	Presentation format is adequate. Showed some confidence & composure but has room for improvement. Presentation may sometimes not consider the needs of the audience. Weak demonstration of the system.	Showed no confidence or composure. presentation format is not used adequately, and the needs of the audience are not considered. Poor evidence, or no demonstration of the system.

How to get help

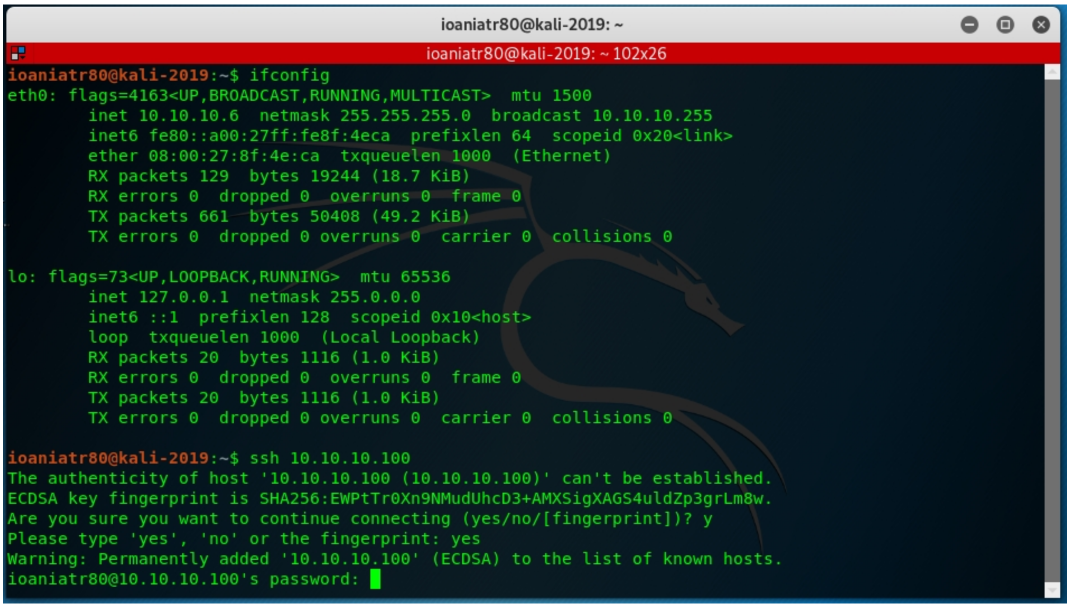
If you have any related questions, please feel free to contact the Module Leader on MS-Teams or via email at iatropoi@lsbu.ac.uk as soon as possible.

Screenshot Requirements Example

The commands to add a new user in a Linux machine are the following:

```
root@raspberrypi:~# useradd -s /bin/bash -m ioaniatr80
root@raspberrypi:~# usermod -aG sudo ioaniatr80
root@raspberrypi:~# passwd ioaniatr80
        New password:
        Retype new password:
        passwd: password updated successfully
root@raspberrypi:~# su ioaniatr80
ioaniatr80@raspberrypi:~$whoami
        ioaniatr80
```

The example below, clearly shows the IP address of this machine by using '*ifconfig*', and '*ssh'* commands.



```
ioaniatr80@kali-2019:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.10.6  netmask 255.255.255.0  broadcast 10.10.10.255
        inet6 fe80::a00:27ff:fe8f:4eca  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:8f:4e:ca  txqueuelen 1000  (Ethernet)
        RX packets 129  bytes 19244 (18.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 661  bytes 50408 (49.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 20  bytes 1116 (1.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 20  bytes 1116 (1.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ioaniatr80@kali-2019:~$ ssh 10.10.10.100
The authenticity of host '10.10.10.100 (10.10.10.100)' can't be established.
ECDSA key fingerprint is SHA256:EWPtTr0Xn9NMudUhcD3+AMXSigXAGS4uldZp3grLm8w.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '10.10.10.100' (ECDSA) to the list of known hosts.
ioaniatr80@10.10.10.100's password: 
```

Figure 1: Screenshot Example (ifconfig execution)

Resources

All the module's lectures, tutorial handouts, and references are recommended in the module guide. Some links that may be found useful:

VirtualBox: Link
Debian-OS: Link
How to install Debian-OS on VirtualBox: Link
DOSBox Emulator: Link
AbandonewareDOS: Link
MyAbandonware: Link
Top 30 Linux Monitoring Tools: Link
How to measure GPU: Link
GLMark2: Link
Geekbench: Link
DOS Benchmark tools Pack: Link

To learn more on how to access research E-resources and publications. Please visit:

https://library.lsbu.ac.uk/friendly.php?s=LSBU-Library-and-Learning-Resources/Home

Quality assurance of coursework specifications

Coursework specifications within CSI division go through internal (for new modules with 100% coursework also through external) moderation. This is to ensure high quality, consistency and appropriateness of the coursework as well as to share best practice within the CSI division.