Northumbria University
NEWCASTLE

COURSEWORK ASSESSMENT SPECIFICATION

Module Title:	Computer networks, security, and operating systems
Module Number:	KV5002
Module Tutor:	Bo Wei
Academic Year:	2019-20
% Weighting (to overall module):	100%
Coursework Title:	Network and Operating Systems Programming (50%) and OS Theory and Concepts (50%)

Dates and Mechanisms for Assessment Submission and Feedback

Date of Handout to Students: week b/w 10/02/2020
Mechanism for Handout to Students: via Blackboard.
Date and Time of Submission by Student: submit by 11:59pm on the 14/5/2020
Mechanism for Submission of Work by Student: via Blackboard.
Date by which Work, Feedback and Marks will be returned to Students: 20 working days after submission deadline.
Mechanism for return of assignment work, feedback and marks to students: via Blackboard/email.

KV5002: Computer networks, security, and operating systems

100% of Module Mark

Feb 2020

Module Tutor: Dr Bo Wei

Ellison Building
Northumbria University
Newcastle-upon-Tyne
NE1 8ST

<bo.wei@northumbria.ac.uk>

Contents

Tasks and Criteria

1 Network and Operating Systems Programming (50%).

1.1 Overview

In the classic Lunar lander games the player controls the descent of a Lunar Lander spacecraft onto the surface of the moon[1,2].

You are to write a program to control a lunar lander. You will be provided with a server that models the flight dynamics and propulsion of the lander. Your program should communicate with the server via UDP, using the application layer protocol described in section 1.3.1. The purpose of communication with the server is to control the flight of the lander. Your program should also communicate with a dashboard display to show the current status of the lander. Finally, the program should write to a data logging file to record the inputs from the user and the response of the lander. Figure 1 shows an overview of the system.

Skeleton codes are provided. Skeleton codes are fully functional with all important elements, but the implementation of some parts is missing. Students are expected to complete these elements for this program. Please note students do not have to write the program based on the given skeleton codes.

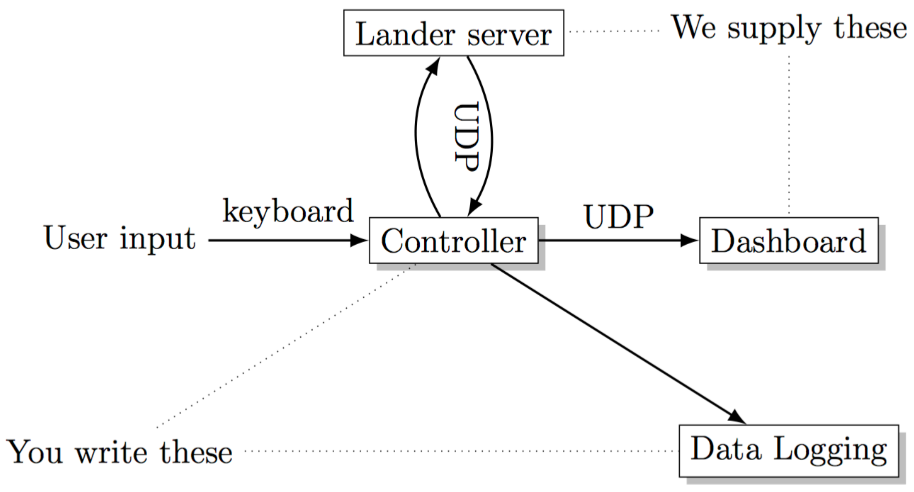

Figure 1: Architecture of the lunar lander software

1.2 Controller

1.2.1 Inputs

Using the keyboard of your PC, accept inputs to control the throttle and the strength of the rotational thrust of the lander. You may choose the mapping of input keys to control signals for yourself. This should be documented in your report.

1.2.2 Communications

The controller needs to manage its communications with the Lander server and the dashboard. Communications with the Lander server are by UDP Datagrams, the format of the messages and responses are given in section 1.3.1. Communications with the dashboard should also use UDP. You should design and implement your own protocol for this purpose. This should be documented in your report.

1.2.3 Control Logic

The controller needs to take the requested actions from the user, and the current state of the lander. From these it should generate the control signals to send to the lander server and the outputs presented to the dashboard. It should also log data to a text file. You should organise your program using at least 4 POSIX threads: user input, server communications, dashboard communications, and data logging. Any data shared by these threads should be protected, as required, by the use of semaphores.

1.3 Lander Server

The lander server is a Java program that models the dynamics of the lander and the lunar surface. You should clone the repository at https://github.com/weibo053/LunarLander (also available in the blackboard under the Assessment and Submission folder) and follow the instructions to build the program. The lander server is packaged as a jar file and executed using

```
java -jar LunarLander.jar
```

1.3.1 Communications Protocol

The server listens for UDP Packets.
Messages to the server, and replies from the server, are formatted as Key:Value pairs, as is done in email headers (RFC822[3]) or HTTP Headers (RFC7230[4]).

Query messages have the message name as the first key, with a single question mark (ascii 63) as the value.

Command messages that set values or cause actions, have the message name as the first key and a corresponding exclamation mark (ascii 33) as the value.

Reply messages, returned in response to the above, have the message name as the key and an equals sign (ascii 61) as the value.

Request State This message requests the current state of the Lander:

Message send

```
state:?
```

Reply from server

```
state:=
x:45.6
y:10.3
O:5
x':0.1
y':-1
O':+0.01
```

The keys and values are:

x	x position
y	y position
O	θ orientation (capital O)
x'	$\dot{x}$ horizontal velocity (x single-quote(ascii 39))
y'	$\dot{y}$ vertical velocity (y single-quote(ascii 39))
O'	$\dot{\theta}$ rate of rotation (capital O single-quote(ascii 39)))

Request condition This message requests the general condition of the Lander:

Message send

```
condition:?
```

Reply from server

```
condition:=
fuel:98%
altitude: 20.7
contact:flying|down|crashed
```

The keys and values are:

`fuel`	percentage of fuel remaining	
`altitude`	the radar altitude above the ground	
`contact`	A keyword describing the general state of the Lander	
	`flying`	moving above the terrain
	`down`	in contact with the ground in a successful landing
	`crashed`	in contact or underground in a failed landing

Command Engines This message sets the requested levels on the engines

Message send

```
command:!
main-engine: 100
rcs-roll:  +0.5
```

The keys and values are:

`main-engine`	percentage throttle setting	0...100
`rcs-roll`	the strength of the rotational thrust +ve is anti-clockwise	-1...1,

Request Terrain This message requests the terrain below the lander from its mapping radar

Message send

```
terrain:?
```

Reply from server

```
tarrain:=
points:10
data-x: 10.0,15.0,20.0,...
data-y: 12.4,12.7,13,14.0,...
```

The keys and values are:

`points`	The number of data points
`data-x`	an array of x ground coordinates
`data-y`	an array of y ground coordinates

The ground coordinates are supplied one (x, y) pair per line of text. The coordinates are absolute and measured in the same system as the lander state.
If the Lander's (x, y) position matches a ground (x, y) position, it has reached that point on the ground. The altitude reported should be the difference between y coordinates of the lander and the ground directly below it.

1.4 Game Dashboard

You are provided with a limited interface on the server, mainly for monitoring the server and debugging. The dashboard provides a more detailed user interface than can be provided by the server.

1.4.1 Requirements

The Dashboard should run on a PC or other computer connected to the controller via a network. You should clone the repository at https://github.com/weibo053/LanderDash (also available in the blackboard under the Assessment and Submission folder) and follow the instructions to build the program. The dashboard can be executed using the command:

```
java -jar LanderDash.jar
```

The dashboard accepts UDP datagrams as input. You should send messages to the dashboard to maintain the current status of all dashboard controls and your protocol should be able to support this requirement.

1.5 Skeleton codes

If you use the skeleton codes, you should get the code in the blackboard under the Assessment and Submission folder and follow the instructions to build the program. The skeleton codes can be compiled by using the following command:

```
make all
```

The controller can be executed using the command:

```
./controller 65200  65250
```

1.6 Additional requirements

If you use the skeleton codes, you should follow the instruction in Section 1.5. You must include your code for controller.c and libnet.c as appendix 1 in your report. If you do not use the skeleton code, you must include your own codes in your report.

Please compress all your codes to an archive file (such as a zip file) and upload it to the blackboard Assessment -> Program. Please ensure it is possible to build and run the controller with the lander server and dashboard, exactly as described earlier. If you do not use the skeleton code, you must include a Makefile.

Failure to satisfy one or more of these requirements will lead automatically to the award of a mark of 0 for the program as described in section 1.7

1.7 Marking of the program

Your program will be assessed on:
1. the extent to which it satisfies the specified functional requirements

(a) interaction with the user via the keyboard	(2 marks)
(b) communication with, and control of, the lander	(4 marks)
(c) updating of the dashboard	(4 marks)
(d) data logging	(2 marks)

2. the quality of the code:

(a) When built with the command make all, your program should compile without errors or warnings. Please note that your program is expected to have all specified functional requirements listed above (8 marks)
(b) Correct use of threads, semaphores, and network APIs; functional decomposition, data structures, layout, naming etc. (10 marks)

1.8 What to include about the program in the report
The first section of your report should be entitled 'The lunar lander controller' and should include the following:
(a) A discussion of the software architecture of the program, focusing in particular on the use of threads and semaphores. What are the advantages and disadvantages of this architecture in fulfilling the functional requirements of the program? How has the program managed user input (6 marks)
(b) A description and critical evaluation of the protocol that the program has used to communicate with the dashboard. To what extent does the protocol help to ensure that communication is reliable? (7 marks)
(c) A description of the data logging that you have planned. What data should the program log and why? What type of file should the program use to store the data? Why should you choose this type? How often should the program log data? Justify this decision. At what rate does the size of the file grow while the lander is being controlled? How have you determined this? (7 marks)

These points are expected to be discussed based on one program. However, you can still discuss these points in your report based on your understanding of the provided skeleton code.

1.9 Summary
You have to include the following three parts for Network and Operating Systems Programming (50%) in your submission:
(1) Please compress all your codes to an archive file (such as a zip file) and upload it to the blackboard Assessment -> Program. More details can be found in Section 1.6.
(2) You must include your codes as appendix 1 in your report. More details can be found in Section 1.6.
(3) The first section entitled 'The lunar lander controller'. More details can be found in Section 1.8.

2 OS theory and concepts (50%)

Sections 2 and 3 of your report should include the following:

Section 2 OS abstraction and process management
(a) Please list three advantages of command line in Linux Operating Systems and give one example for each advantage. (9 marks)
(b) Please describe the difference between the processes and threads. (8 marks)
(c) Describe in detail the actions taken by an operating system to achieve a context switch between processes. Illustrate your answer with diagrams. Please draw diagrams according to your own understanding and do not copy them from other sources. (8 marks)

Section 3 Security
An important requirement of many operating systems is to provide a secure communication function between processes. One approach to the provision of such a function is the Secure Socket Layer (SSL).
(a) Identify two potential security violations are addressed by SSL. Explain briefly how SSL offers protection against each potential violation that you identify.   (8 marks)

(b) SSL makes use of both symmetric and public-key cryptography. Explain what you understand by these concepts, distinguishing clearly between them. Give examples of applications of each cryptography method to show how each of these cryptographic techniques is used in SSL, explaining the reasons for the choice of technique in each case. (8 marks)
(c) SSL is susceptible to a man-in-the-middle attack. Explain the nature of this attack. Illustrate your answer with one diagram. Identify the precise vulnerability of SSL to this attack and discuss how users of SSL can protect themselves against it. (9 marks)

Reference:
[1] Lunar Lander. Wikipedia. [Online]. [Accessed 2 Dec 2019]
https://en.wikipedia.org/wiki/Lunar_Lander_(video_game_genre)
[2] Lunar Lander. Wikipedia. [Online]. [Accessed 2 Dec 2019]
https://en.wikipedia.org/wiki/Lunar_Lander_(1979_video_game)
[3] RFC822. IETF. [Online]. [Accessed 2 Dec 2019]. https://tools.ietf.org/html/rfc822
[4] RFC7230. IETF. [Online]. [Accessed 2 Dec 2019]. https://tools.ietf.org/html/rfc7230

Learning Outcomes assessed in this assessment

1. Demonstrate knowledge and critical understanding of networking fundamentals, including network architecture and protocols.
2. Demonstrate knowledge and critical understanding of the fundamentals of an operating system, including its architecture and the implementation of its services.
3. Demonstrate knowledge and critical understanding of operating systems and network security, including fundamental concepts, threats, attacks and basic techniques for defence.
4. Interpret a requirements specification to design and develop a networked application program, using standard operating systems and networking APIs (e.g. POSIX.1-2008), and demonstrating cognisance of security issues and industry best practice.
5. Communicate the results of work/study reliably and accurately.

Marking Scheme

This assignment assesses all the module learning outcomes specified above. In writing your report, please bear in mind the guidance given in the notes above on what is expected, and the criteria associated with each task in the marking criteria below.

1. a software development project assessing systems programming and concurrency (50%)
2. questions on OS theory and concepts (50%)

More detailed marks allocation is provided in Section Tasks and Criteria and Section Assessment Criteria/Grade Descriptors.

Submission Format/Requirements

Your report should be submitted using Turnitin UK via Blackboard under 'Assessment'. Your codes should be submitted using Blackboard under 'Assessment -> Program'.

For the report:

- All content including references and appendices should be contained in a **single document**.

- **Only Microsoft Word (.docx) file formats** will be accepted.
    - The report should be written in a **formal and professional reporting style**
- The report should be written **in the third person without use of personal pronouns** (for example, no use of 'I, me, my, our, we, they'). If you find this difficult, try using a passive voice.
- Layout should make reasonable use of margins, clear headings and font size should be 10-12pt (depending on choice of font), preferably using a Sans-Serif typeface although you may use a Serif-based font should you wish.
- Expected size of the submission: Your report should be about 5 to 7 A4 pages in length (assuming 10pt, normal margins, and excluding appendices). There is no fixed penalty for exceeding this limit but unnecessary verbosity, irrelevance and 'padding' make it difficult for the marker to identify relevant material and may lead to some loss of marks.
- Please note, these assessments are marked anonymously so do not include your name anywhere on the document.
- Referencing should be in the Harvard style.

Opportunities for Formative and Final Feedback

During each seminar/workshop session, formative discussions will occur on how the learning undertaken can both support and evidence successfully meeting the programme and module learning outcomes. The seminars/workshops will also offer opportunities for formative feedback.

Unmoderated marks and summative feedback for the assessment will be returned within four working weeks after the final submission deadline.

Please use the classes for clarification, guidance or support on any aspect relating to this assessment.

Academic Integrity

You must adhere to the university regulations on academic conduct. Formal inquiry proceedings will be instigated if there is any suspicion of plagiarism or any other form of misconduct in your work. Refer to the University's Assessment Regulations for Northumbria Taught Awards if you are unclear as to the meaning of these terms. The latest copy is available on the University website.

- https://www.northumbria.ac.uk/about-us/university-services/academic-registry/quality-and-teaching-excellence/assessment/guidance-for-students/

Remember, this is **INDIVIDUAL** assessment and should be entirely your own work. Where you have used someone else's words (quotations), they should be correctly quoted and referenced in accordance to the Harvard System.

Failure to submit: The University requires all students to submit assessed coursework by the deadline stated in the assessment brief. Where coursework is submitted without approval after the published hand-in deadline, penalties will be applied as defined in the University Policy on the Late Submission of Work; please refer to the link below.

- https://www.northumbria.ac.uk/about-us/university-services/academic-registry/quality-and-teaching-excellence/assessment/guidance-for-students/

Anonymous marking: University policy requires that work be marked anonymously. In order to facilitate this, we request that only your student number is included on work submitted

Academic Misconduct Policy:

- https://www.northumbria.ac.uk/about-us/university-services/academic-registry/quality-and-teaching-excellence/assessment/guidance-for-students/

Assessment Criteria/Grade Descriptors

Generic marking criteria for the report

Mark	**Generic Assessment Criteria**
70%+	Excellent work providing evidence to a very high level of the knowledge, understanding, and skills appropriate to Level 5. All learning outcomes met, many at an excellent level. Marks at the high end of this range indicate outstanding work where all learning outcomes are met at an excellent level. Excellent in all (or most of): relevant, appropriate, accurate, effectively-communicated, and complete discussion; cohesive assessment, representation and evaluation of current issues and trends; arguments are sound, well justified, and well presented; technical and professional understanding; use of high-quality up-to-date information sources; evidence of independent learning.
60-69%	High-quality work providing evidence to a high level of the knowledge, understanding, and skills appropriate to Level 5. All learning outcomes met, many are more than satisfied. Very good in all or most of: relevant, appropriate, accurate, effectively-communicated, and complete discussion; cohesive assessment, representation and evaluation of current issues and trends; arguments are sound, well justified and well presented; technical and professional understanding; use of high-quality information sources; evidence of independent learning.
50-59%	Satisfactory work providing evidence of the knowledge, understanding, and skills appropriate to Level 5. All learning outcomes met. Satisfactory in all or most of: relevant, appropriate, accurate, and effectively-communicated discussion; cohesive assessment, representation and evaluation of current issues and trends; arguments are presented; technical and professional understanding; use of external information sources; evidence of independent learning.
40-49%	Adequate work providing evidence of the knowledge, understanding, and skills appropriate to Level 5 but at a bare pass level. All learning outcomes are met (or nearly met and balanced by strengths elsewhere). Adequate in all of (or most of, with balancing strength elsewhere): relevant, appropriate, accurate, and effectively-communicated discussion; cohesive assessment, representation and evaluation of current issues and trends; arguments are presented; technical and professional understanding; use of external information sources; evidence of independent learning.

0-39%	Work is not acceptable in providing evidence of the knowledge, understanding and skills appropriate to level 5. Only some of the learning outcomes are met. Inadequate in some of the following aspects or seriously inadequate in at least one: relevant, appropriate, accurate, and effectively-communicated discussion; cohesive assessment, representation and evaluation of current issues and trends; arguments are presented; technical and professional understanding; use of external information sources; evidence of independent learning. OR Work not submitted. OR Work shows no evidence of the knowledge, understanding and skills appropriate to level 5. None of the learning outcomes are met OR Work gives evidence of serious academic misconduct.