

# ICT374 Operating Systems and Systems Programming (tma, 2020)

## Lab Exercises

### ICT374 Lab 11:

### Specification of Network Protocol, Checking Open Ports on a Host with netstat, Filename Matching Using glob Function, Communicating with HTTP Server Using telnet Client, Virtual Memory Based on Paging.

#### Learning Objectives:

1. Learn how to write a protocol specification;
2. Learn how to check ports open on a Unix host with netstat;
3. Learn to handle wildcards in filenames using glob function;
4. Learn to exchange HTTP messages with HTTP servers using telnet client (or the command nc if the command telnet is not available in your machine);
5. Understand virtual memory based on paging.

#### Required Reading:

- Lecture Notes for Topic 8
- Rute: Ch 4.3 (glob)
- Rute: Ch 26.2 (telnet web server)
- Stallings: Ch 7 & 8. (virtual memory)

#### Exercises

##### 1. Specifying a network protocol - an example

This exercise is relevant to Project - A Simple File Transfer Protocol.

A network protocol must specify all three of the following:

- a. the format of messages exchanged over the network
- b. the content of the messages exchanged over the network
- c. the sequence of messages exchanged over the network

Let's use the example of sending a file from the client to the server (which is a part of Project - A Simple File Transfer Protocol).

a. Client sends one message to the server. The message format is given below:

- one byte opcode which is ASCII character A followed by
- two-byte integer value in two's complement and in network byte order, which represents the length of the filename of the file to be sent to the server followed by
- the sequence of ASCII characters representing the filename

b. Once the server receives a message with opcode A, it responds with the following message:

- one byte opcode which is ASCII character A followed by
- one byte acknowledgment code which is one of the following ASCII characters:
  - 0 - the server is ready to accept the named file
  - 1 - the server cannot accept the file as there is a filename clash
  - 2 - the server cannot accept the file because it cannot create the named file
  - 3 - the server cannot accept the file due to other reasons.

c. Once the client receives the A message from the server and if the acknowledgment code is 0, it responds by sending the following message:

- one byte opcode which is the ASCII character B followed by
- four-byte integer (let's call it N) in two's complement and in network byte order which represents the length of the file followed by
- a sequence of N bytes which is the content of the file

Note in the above specification, it is preferable to use diagrams to help illustrate the message format (I have not drawn the diagrams because it is difficult to do it using HTML). Also note: the above specified the message format (such as 1 byte, 2 bytes,



Note also in the above specification, we did not say how the client should report the error message to its user. The latter is not part of this protocol. Here it is an implementation issue which should be left to the implementor of the client to decide. For example, the client implementation may report the error as a text message in English, or in Chinese, or as an image on the desktop, or does not report the error at all.

In the above specification, words like "ASCII character A" are necessary. Imagine if the words were instead "character A", it would open up for many possible interpretations (such as EBCDIC character, or UTF-16 character etc). The same applies to words like "four-byte integer in two's compliment and in network byte order". If you miss any one word, the meaning will be ambiguous.

2. Specifying a network protocol - an exercise

This exercise is relevant to Project - A Simple File Transfer Protocol.

Now it is your turn. Write a protocol specification that specifies how the client downloads a file from the server.

3. Checking ports opened on a Unix host with netstat

This exercise is useful when you test network programs. It allows you to find out which port is already used.

The command netstat allows you to see the ports, such as TCP and UDP ports, that are open on a Unix host and their current status, such as listening port, or whether a connection is established with a foreign host.

Try the following command to see all TCP ports open on a Linux host:

```
netstat -a | grep tcp | more
```

Identify the connections between the local host and the remote foreign hosts. Identify the listening ports.

4. Using glob function to expand wildcards in filenames

This exercise is relevant to the Project - A Simple Unix Shell. You can use the function glob to handle the file names containing wildcard characters "\*" and "?".

Read the manual page for glob:

```
man glob
```

and read the following web page for the glob function:

<http://www.die.net/doc/linux/man/man3/glob.3.html>

Write a C program that takes a filename pattern containing wildcards "\*" and "?". Your program should print out all filenames that matches the pattern.

5. Using telnet to see HTTP messages

This exercise is relevant to Project 3: A Simple HTTP server. From this exercise you can find out what is actual exchanged between the web server and web browser.

Read Ch 26.2 of Rute. Follow the instructions to manually exchange messages with a web server using the telnet client.

From this exercise you should get a feeling of what is going on between a web server and a web browser. The understanding of this will help you design your web server.

If the command telnet is not installed on your machine, you may use the command nc to achieve the same purpose. Check the manual page for nc.

6. Virtual memory based on paging

A simple virtual memory system has 32KB physical memory with 16-bit virtual address, of which 12 bits are used as offset. The following is the current content of the page table of one of the processes: given below:

Virtual Page Number	Page Frame	Bits Present / Absent bit
15	000	0
14	000	0
13	000	0
12	000	0
11	111	1
10	000	0
9	101	1
8	000	0
7	000	0
6	000	0
5	011	1
4	100	1
3	000	1
2	110	1
1	001	1
0	010	1



- a. What is the page size of this virtual memory system?
- b. What is the corresponding physical address of virtual address B2A0 (in hexadecimal notation)?
- c. What would happen if the CPU executes an instruction to move an integer from memory address A610 (hexadecimal) to a register?
- d. What are the advantages and disadvantages of keeping the page table in the main memory?
- e. Describe a realistic page replacement algorithm. Note that you may need to augment the above page table with additional attributes to suit the needs of your algorithm.

Lab Assignment:

Although this lab is not assessable, it is highly recommended that you complete all exercises in this lab. It contains useful information for your second assignment particularly if your assignment involves the design of network protocols. The exercises and questions in this lab could also be relevant to your final examination.

- 1. Complete Exercise 2.
- 2. Complete Exercise 3.
- 3. Complete Exercise 6.

Dr Hong Xie  
Unit Coordinator

Last update: Tuesday, 24 April 2018

Document author: [Dr. H. Xie](#), Unit Coordinator, ICT374 Operating Systems and Systems Programming  
Last Modified: Tuesday, 24-Apr-2018 14:08:28 AWST  
[Disclaimer & Copyright Notice](#) © 2020 [Murdoch University](#).  
This document is relevant for 2020 only



Last modified: Thursday, 2 August 2018, 1:09 PM

[◀ Lecture Notes](#)

Jump to...

⌵

[Assignments ▶](#)

