Preview Test: CSSE2310 Semester Two Final Examination 2020 - PART A

Test Information

Description Undertaking this online examination deems your commitment to UQ's academic integrity pledge as summarised in the following declaration:
"I certify that I have completed this examination in an honest, fair and trustworthy manner, that my submitted answers are entirely my own work, and that I have neither given nor received any unauthorised assistance on this examination".

Instructions There are 75 marks for this exam. Answers all questions except the final coding question within this Blackboard Test. Your files for the final coding question must be uploaded to the Blackboard assignment item found on the Blackboard exam page. This upload must be completed prior to the submission deadline for this test. **Do NOT submit this Blackboard test until <u>after</u> you have uploaded the files - you will lose access to the specification for the coding question when this test is submitted.**

Timed Test This test has a time limit of 2 hours and 30 minutes.This test will save and be submitted automatically when the time expires.
Warnings appear when **half the time**, **5 minutes**, **1 minute**, and **30 seconds** remain.
[The timer does not appear when previewing this test]

Multiple Attempts Not allowed. This test can only be taken once.

Force Completion This test can be saved and resumed at any point until the time has expired. The timer will continue to run if you leave the test.

Your answers are saved automatically.

QUESTION 1   **1 points** | Save Answer |

Write a shell command to show the names (only) of all files and directories in the "data" subdirectory of the /tmp directory (which is NOT the current directory)

[   ]

QUESTION 2   **1 points** | Save Answer |

Write a shell command to copy all .c files in the /tmp directory to the csse2310 subdirectory of the current directory

[   ]

QUESTION 3

1 points   Save Answer

Write a shell command to make a new subdirectory of the current directory called "orange"

[   ]

QUESTION 4

1 points   Save Answer

Write a shell command to create an executable program called `doit` that uses pthreads from a C file called `docalc.c`. All input and output files are in the current working directory. (Do <u>not</u> include any unnecessary C compiler arguments, e.g. -pedantic etc.)

[   ]

QUESTION 5

1 points   Save Answer

Write a shell command to bring any changes from the subversion repository into the working directory (the current directory).

[   ]

QUESTION 6

1 points   Save Answer

Write a shell command to show all lines in a file called "animals" (in the current directory) that contain the string of characters "dog".

[   ]

QUESTION 7

1 points   Save Answer

Write a shell command to show all lines in a file called "animals" (in the current directory) that do not contain the word "possum".

[   ]

QUESTION 8

1 points   Save Answer

Write a shell command that finds all lines in the file **netconfig** (in the **/etc** directory) that contain the string "inet" (without the quotes) but not string "inet6" (without the quotes) and appends those lines to a file called **nets** in the **tmp** subdirectory of the current directory.

[   ]

QUESTION 9

1 points   Save Answer

Write a shell command that counts the number of lines in a file called "addresses" (in the current directory) that contain the word "Toowong".

[   ]

QUESTION 10

1 points   Save Answer

Write a shell command that counts all lines in the file **nsswitch.conf** (in the **/etc** directory) that contain the word "ldap" (without the quotes) and writes that count value to a file called **ldap.count** in the **tmp** subdirectory of the current directory.

[   ]

QUESTION 11

1 points   Save Answer

Write a shell command that, in the current directory, creates a symbolic link called "apple" that points to "mac"

[   ]

QUESTION 12

Write C declarations to declare foo as ...
(a) An array of seven non-negative whole numbers

[   ]

(b) A pointer to a function which takes three integers as parameters
and returns a string [   ]

(c) A character which is modified by multiple threads

[   ]

(d) A pointer to a high precision floating point number

[   ]

(e) An array of three true/false values [   ]

(f) A pointer to a function that is suitable for use as a signal handler

[   ]

Consider a system with 8KiB pages and the following page table. All numbers are in base 10.

Page Number	Frame Number
0	-
1	30
2	-
3	31
4	32
5	23
...	
20	21
21	20
22	43
23	112
...	
48,325	132
48,326	99

For each of the following virtual addresses, what is the corresponding physical address? Write your answer in base 10. If accessing the virtual address would result in a segmentation fault, then write "SEGFAULT". If there is insufficient information in the page table (i.e. the page number is not listed) then write "UNKNOWN".

8,192:

20,480:

48,326:

86,123:

QUESTION 14

6 points

Suppose a system uses 39-bit virtual addresses, 48-bit physical addresses and a three level page table. Pages are 4KiB in size. Page table entries are 8 bytes each.

A process uses the following virtual address range (all numbers are in base 10):

- 1200 MiB starting at address 0

Enter your answers below as decimal numbers. Do not include the units in your answer.

(i) What is the maximum memory size for a process (in GiB)?

[_____] GiB

(ii) How much memory (in KiB) would be needed to store the page table? [_____] KiB

(iii) If the process doubled its memory usage (to 2400 MiB starting at address 0), how much memory (in KiB) would now be needed to store the page table? [_____] KiB

(iv) If the original process doubled its memory usage (to 2400MiB - 1200MiB at address 0, and 1200MiB at the highest possible addresses), how much memory (in KiB) would now be needed to store the page table? [_____] KiB

(v) For the original memory usage (1200MiB starting at address 0), if the system used a two level page table, how much memory (in KiB) would be needed to store the page table? [_____] KiB

(vi) If the system used a single level page table, how much memory (in MiB) would be needed to store the page table?

[_____] MiB

Consider the following program:

```c
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>

int main(int argc, char** argv) {
    fprintf(stderr, "A");
    if(fork()) {
        fprintf(stderr, "B");
    } else {
        printf("C");
        if(fork()) {
            fprintf(stderr, "D");
        } else {
            fprintf(stdout, "E");
        }
    }
    fflush(stderr);
    fflush(stdout);
    fork();
    fprintf(stderr, "F");
    return 0;
}
```

Assuming that output to stderr is never buffered, which of the following statements is true?

○ 'C' may appear twice in the output

○ An 'F' can only appear in the output if a 'B' has already appeared.

○ The last character output must always be an 'F'

○ A 'D' can only appear in the output if a 'C' has already appeared.

○ An 'F' can only appear in the output if a 'D' has already appeared.

5 points   Save Answer

Consider the following program. Assume that all system calls succeed.

```c
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>

int main(int argc, char** argv) {
    printf("A\n");
    if(fork()) {
        printf("B\n");
        fork();
    } else {
        printf("C\n");
        if(fork()) {
            printf("D\n");
            wait(NULL);
        } else {
            printf("E\n");
        }
    }
    fork();
    printf("F\n");
    return 0;
}
```

How many processes are created by the execution of this program (including the initial process)? **8**

What is the maximum number of processes that could be running at any point? **7**

If standard output buffers are never flushed until a process exits, how many times will the letter C be output? **4**

What is the minimum number of lines of text that this program will output when run? **13**

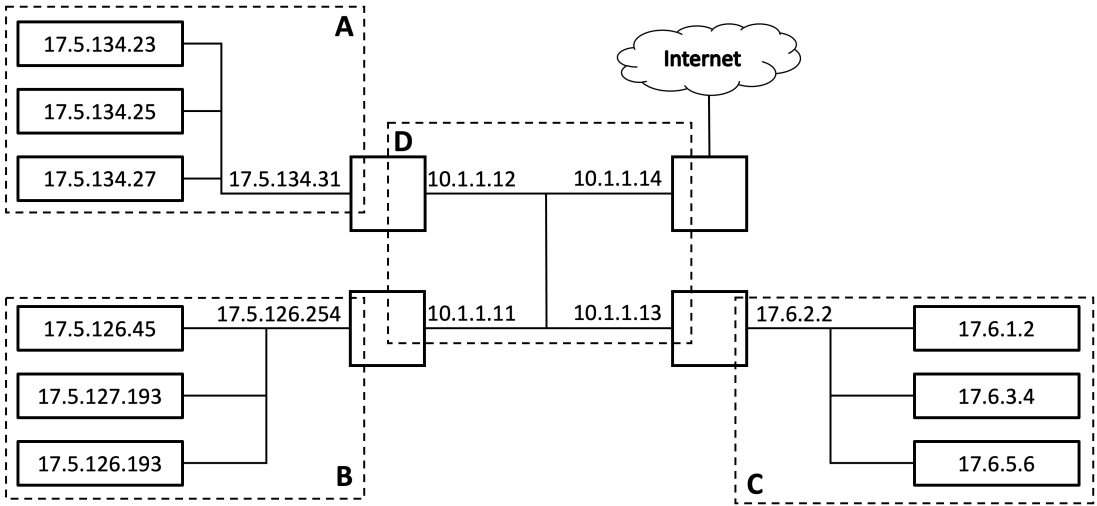What is the maximum number of lines of text that this program will output when run? **28**

Consider the following network (assume that all networks are as small as possible):

A
17.5.134.23
17.5.134.25
17.5.134.27   17.5.134.31

D
10.1.1.12   10.1.1.14

Internet

17.5.126.254   10.1.1.11   10.1.1.13   17.6.2.2

17.5.126.45
17.5.127.193
17.5.126.193   **B**

C

17.6.1.2
17.6.3.4
17.6.5.6

Fill in the netmask, broadcast address and CIDR for each of the networks (A to D):

Network	Netmask	Broadcast Address	CIDR
A			
B			
C			
D			

Fill in the detail for the whole network shown above (as it would appear to the rest of the internet).

Netmask	Broadcast Address	CIDR

How many unused addresses are there in network A? (i.e. how many additional machines could be added to the network)

QUESTION 18

2 points   Save Answer

Consider a "unix" file system where

- blocks are 4 KiB
- block pointers are 8 bytes
- inodes have
  - 7 direct pointers
  - 3 single indirect pointers
  - 2 double indirect pointers

What is the maximum file size on this file system? Express your answer in <u>bytes</u>.

[   ]

QUESTION 19

2 points   Save Answer

Consider a "unix" file system where

- blocks are 8 KiB
- block pointers are 8 bytes
- inodes have
  - 6 direct pointers
  - 4 single indirect pointers
  - 3 double indirect pointers

What is the largest file size that can be stored <u>without</u> using a double indirect pointer? Express your answer in KiB.

[   ]

QUESTION 20

1 points   Save Answer

Consider a "unix" file system where

- blocks are 16 KiB
- block pointers are 8 bytes
- inodes have
  - 6 direct pointers
  - 4 single indirect pointers
  - 2 double indirect pointers

Assuming the inode is cached in RAM, how many blocks would need to be accessed to read byte number 325,383 in the file?

[   ]

Consider the following directory listing:

```
$ ls -ali
total 2164
5067385 drwxrwxr-x   3 alice       staff     4096 Oct 1
7 11:08 .
  75112 drwxr-xr-x 100 root        root      4096 Oct 1
7 10:43 ..
5061551 -rwxr-xr-x   1 bob         staff     8192 Oct 1
8 10:23 admin
5067391 lrwxrwxrwx   1 bob         users        4 Oct 1
7 10:59 backup -> data
5066833 drwxr-x---   5 bob         staff     4096 Oct 1
7 10:59 data
5067386 -rwxr-xr-x   1 alice       staff 2190232 Oct 1
7 11:00 generate
5068104 lrwxrwxrwx   1 alice       staff        8 Oct 1
7 11:09 install -> generate
5067407 -rw-r--r--   1 bob         staff      279 Oct 1
7 11:05 file2
5067392 -rw-r--r--   1 alice       staff      279 Oct 1
8 11:05 files
```

Users "alice" and "bob" are the only members of the "staff" group. All users on the system are members of the "users" group.

Within this directory, what command can alice run to prevent bob from being able to run **generate** but still allow other users of the

system to run that program? [   ]

Within this directory, what command can bob run to allow all users other than alice to list the contents of the **data** subdirectory?

[   ]

How many subdirectories does the **data** subdirectory have?

[   ]

At some later point in time, the same command (**ls -ali**) includes two additional lines:

```
5067407 -rw-r--r--   2 bob         staff      279 Oct 1
7 11:05 filex
5067408 lrwxrwxrwx   1 alice       users        4 Oct 1
9 11:08 filez -> files
```

The contents of **file2**, **files** and **filex** are identical.

What command could have been used to create **filex**?

[   ]

What command could have been used to create **filez**?

[   ]

3 points Save Answer

A system has the following ordinary users and groups (and no others):

User	Groups
alice	staff, users, project
bob	staff, users
carol	admin, users
dave	admin, users, project
eve	project, users

Consider the following directory listing:

```
-r--r-x--- 1 alice staff    138856 Oct 17 10:46 file1
--w-r-xrwx 1 dave  project 2190232 Oct 17 11:49 file2
---xr--r-x 1 eve   project   24000 Oct 17 09:14 file3
```

Which users are allowed to do the following? Enter your answer as a comma separated list of usernames in alphabetical order.

1. Read from file1 [   ]

2. Write to file 2 [   ]

3. Run file3 [   ]

Consider a file **listen.c** that has the following contents:

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netdb.h>
#include <string.h>

int listen_on_port(char* port) {
    struct addrinfo* ai = 0;
    struct addrinfo hints;
    memset(&hints, 0, sizeof(struct addrinfo));
    hints.ai_family = AF_INET;
    hints.ai_socktype = SOCK_STREAM;
    getaddrinfo("localhost", port, &hints, &ai);
    int server_fd = socket(AF_INET, SOCK_STREAM,
0);
    bind(server_fd, (struct sockaddr*)ai->ai_addr,
sizeof(struct sockaddr));
    listen(server_fd, 1);
    return server_fd;
}
```

and the file **listen.h** that contains the following function prototype:

```
int listen_on_port(char* port);
```

For your convenience, copies of these files can be found on **moss.labs.eait.uq.edu.au** within the directory `/local/courses/cs se2310/resources/exam`

Write and upload a file called **netexec.c** (12 marks) and an associated **Makefile** (5 marks) that will create an executable **netexec** that has the following behaviour.

./netexec *port1* *port2* *prog* *arg1* *arg2* ...

The program will listen on both *port1* and *port2* and will wait for a single connection to each port (in any order) and then run **prog arg1 arg2 ...**. (There may be any number of arguments after the program name, including zero.) Input to the running program must be taken from the connection to the first named port. Output from the running program must be sent to the connection to the second named port. Your program must make use of the `listen_on_port()` function in **listen.c**. Your program must check that sufficient command line arguments are supplied, and if not, then print a message to standard error and exit with a non-zero exit status. No other errors need to be checked for. You may assume that all system calls succeed. Note that input and output may be binary (i.e. not just lines of text).

An example execution of netexec may be as follows: **./netexec 43200 55123 grep hello**

This will listen on ports 43200 and 55123. If the connection to the first port (43200 in this example) sent

```
abc
hello there
csse2310
```

then the connection to the second port would receive

```
hello there
```

Your program must build and run on moss.labs.eait.uq.edu.au for marking purposes. (You can develop it elsewhere if you wish, but testing will take place on moss.) The maximum mark you can achieve for this question if your code does not compile on moss is 70%. In the absence of a Makefile that builds your program, we will attempt to

build it with the command: `gcc -std=gnu99 -o netexec netexec.c listen.c`

Your **Makefile** must have the following characteristics:

- Compilation and linking must be separate steps, i.e. **netexec** is built from object files.
- Compilation must include the C compiler flags `-std=gnu99` and `-pedantic`. (Others can be included if you wish.)
- (Re)compilation of a C file will only happen if it or a local header file it includes has been modified since the last time it was compiled
- Running `make` without any arguments will build **netexec** (if required)
- Running `make clean` will remove **netexec** and all object files

Your submission (to the Blackboard submission link) must include two files - named **netexec.c** and **Makefile**. Do NOT upload a zip file or any other files.

Indicate below the number of files that you have uploaded. **Do NOT submit this test until _after_ you have uploaded the files - you will lose access to this question detail.**

QUESTION 24

0 points   Save Answer

Please use this space to specify any assumptions you have made in completing the exam and which questions those assumptions relate to. You may also include queries you may have made with respect to a particular question, should you have been able to 'raise your hand' in an examination room.

For the toolbar, press ALT+F10 (PC) or ALT+FN+F10 (Mac).

B	*I*	U	S	Paragraph ∨	Arial ∨

10pt ∨   ≔ ∨   ≔ ∨   A ∨   🖊 ∨   I×

✂ 🗗 📋 🔍   ↶ ↷   •••

≡ ≡ ≡ ≡   →≣ ≣←   x² X₂   🔗 🔗   ›¶ ¶‹   — ±

ABC✓ ∨   √   ¶   "   Ω   ☺   🔖   ⊞ ∨   ⊠   ⊞ ⊞

⊞ ⊞ ⊞ ⊠   ⊞ ⊞ ⊠   <>   🛉   {;}   ⊙   ⑦   ⤢   ⊕

P   0 WORDS   POWERED BY TINY