THE UNIVERSITY OF QUEENSLAND AUSTRALIA

This exam paper must not be removed from the venue

Venue	____________________
Seat Number	________
Student Number	|__|__|__|__|__|__|__|__|
Family Name	______________________
First Name	______________________

School of Information Technology and Electrical Engineering

EXAMINATION

Semester Two Final Examinations, 2016

CSSE2310/7231 Computer Systems Principles and Programming

This paper is for St Lucia Campus students.

Examination Duration:	120 minutes
Reading Time:	10 minutes

Exam Conditions:

This is a Central Examination

This is an Open Book Examination

During reading time - write only on the rough paper provided

This examination paper will be released to the Library

Materials Permitted In The Exam Venue:

(No electronic aids are permitted e.g. laptops, phones)

Calculators - Any calculator permitted - unrestricted

Materials To Be Supplied To Students:

Instructions To Students:

Additional exam materials (eg. answer booklets, rough paper) will be provided upon request.

Attempt all questions.

For Examiner Use Only

Page	Mark

Total ________

Question 1) Write shell commands to do the following: [10 marks (2 each)]

A) Show the 5th line of the file `elephant`.

```
cat elephant | head -5 | tail -1
```

B) Count how many times “mouse” appears in the first column of file `elephant`.

```
cat elephant | cut -d' ' -f 1 | grep mouse | wc -l
```

C) Show all instances of bash which you are running on the system.

```
ps -u $USER | grep bash
```

D) Show the filenames in the current directory which are longer than 8 characters.

```
ls -ad ?????????*
```

E) Add `/srd/bin` to the list of directories to search for commands.

```
PATH=$PATH:/srd/bin
```

Question 2) Write C to declare `foo` as . . . : [5 marks (1 each)]

A) A string which can't be modified.

const char* foo;

B) An integer variable to store large positive values.

unsigned long foo;

C) An array of 12 high precision floating point values.

double foo[12];

D) A pointer to a function which takes three arbitrary pointers and a string and returns a string.

char* (*foo)(void*, void*, void*, char*)

E) A pointer to a function which takes an array of function pointers (each taking a string and returning a string) and returns one of the function pointers it was passed.

(char*) (*(*foo)((char*)*[](char*))) (char*)
{Blue is for foo as pointer to function
With Pink representing it's parameter
The red part is for the return}

Question 3) What is the output from the following statements [10 marks (1 each)]

A)

```
int a=5;
int b=13;
int c;
c=(b^a^b)|b;
printf("%d", c);
```

5: 0101
13: 1101

c = (1101^0101^1101) | 1101
  = 1000^1101 | 1101
  = 0101 | 1101
  = 1101

13

B)

```
int a=26;
int b=9;
int c=6;
int d=a & b | c;
printf("%d",d);
```

26: 011010
9: 1001
6: 0110
d = 011010 & 1001 | 0110
  = 1000 | 0110
  = 1110

14

C)

```
int a=7;
int b=4;
int c=a+b/a*2.5;
printf("%d", c);
```

7: 0111
4: 0100
c = 7+4/7*2.5
  = 7

7

D)

```
int a=1, b=10, c=5;
if ((a--) && (b=5)) {
    c--;
}
printf("%d %d %d", a, b, c);
```

a=0 && b=5
c=4

0 5 4

0 5 4

E)

```
int a=17;
int b=4;
printf("%d", a-b/a+b);
```

17-4/17+4
= 17-0+4 = 21

21

F)

```
float a=3.0;
int b=2;
a/=a+b;
printf("%f", a);
```

a+b=5
a/=5 a=a/5=0.6

0.6

G)

```
int a=17%4;
int b=1;
switch (a) {
    case 0: b=b+2; break;
    case 1: a=2; break;
    case 2: b=b+5;
    case 3: a=a+5;
}
printf("%d %d", a, b);
```

a=17%4=1
b=1

a=2

2 1

2 1

H)

```
int i=5;
int t=0;
for (int i=0;i<9;++i) {
    t=t+i;
}
printf("%d %d", i, t);
```

i=0 0<9 i=0
t=0 i=1

5 36

I)

```
int i=5;
int t=0;
for (i=0;i<10;++i) {
    t=t+i;
}
printf("%d %d", i, t);
```

10 45

J)

```
char* w="fred";
char* x="hw";
char* h=&(x[1]);
w=w+h[1]+1;
printf("%s", w);
```

h="w"
w="red"

red

Question 4) The following system uses 8KB pages (8192Bytes) and a two level page table. [7 marks]
All addresses are given base 10.

A) Suppose process A has the following page table: [2 marks]

Page	**Frame**
0	21
...	...
23	99
24	100
25	19
26	102
27	100
28	101

What physical address correspond to the following virtual addresses:

i) 197493

page = i) 197493 / 8192 = 24.108 = 24
offset = 197493 % 8192 = 885
physical address = 197493 = 100*8192+ 885 = 820085

ii) 222069

page =ii) 222069 / 8192 = 27.108 = 27
offset = 222069 % 8192 = 885
physical address = 222069 = 100*8192 + 885 = 820085

iii) 229376

page = iii) 229376 / 8192 = 28 = 28
offset = 229376 % 8192 = 0
physical address = 229376 = 101*8192 + 0 = 827392

iv) 229375

page = 229375 / 8192 = 27.9998 = 27
offset = 229375 % 8192 = 8191
physical address = 229375 = 100*8192+8191 = 827391

B) Assuming that the TLB starts out empty, how many pages in the page table must be accessed to read from the virtual addresses given above? Assume the reads happen immediately after each other. [2 marks]

i) 2

ii) 2

iii) 1

iv) 1

C) There are two things which are unusual about this page table. What are they, and why are they unusual? [3 marks]

1. two pages map to the same physical address, wasting one page
2. page 0 is a valid page, usually reserved for error checking

Question 5) [8 marks]

Consider the following listings for a filesystem which uses 1KB blocks :

```
/$ ls -l
-rw-r--r--  1 me folk 7763412 May 18 13:31 bsd.rd
drwxr-xr-x 62 me folk    4096 May 18 13:09 Library
drwxr-xr-x  2 me folk    4096 May 18 13:13 tmp
drwxr-xr-x  5 me folk    4096 May 18 13:04 WINDOWS

/$ ls -lih tmp
81920098 lrwxrwxrwx 1 me folk    7 May 18 13:13 fjord -> python3
81920096 -rwxr-xr-x 2 me folk 3.7M May 18 13:11 parrot
81920096 -rwxr-xr-x 2 me folk 3.7M May 18 13:11 perl
81920033 lrwxrwxrwx 1 me folk   19 May 18 13:03 progs -> ../WINDOWS/SYSTEM32
81920097 lrwxrwxrwx 1 me folk    6 May 18 13:12 python -> parrot
81920100 -rw-r--r-- 1 me folk 1.3K May 18 13:39 v2.sh
81920027 -rw-r--r-- 1 me folk  56K May 18 13:37 v.pdf
81920099 -rw-r--r-- 1 me folk 1.3K May 18 13:39 v.sh
```

A) How many (immediate) subdirectories does `/Library` have? [1 mark]

62 - 2 = 60

B) Suppose the above filesystem is full (and no space is reserved), and then

```
rm v.sh v2.sh
```

is executed (in `/tmp`). What is the largest file which could then be created on the filesystem (as a regular user)? [1 mark]

since blocks are 1KB big, 2*ceil(1.3KB) = 4KB

C) What will be the affect of executing the following commands (as `me`):

```
ln parrot python3
rm parrot
```

on these files (and why): [4 marks]

i) fjord

Removing parrot doesn't mean python3 is removed.
So fjord can still access python3 meaning it is unaffected.

ii) python

Dangling soft link

iii) perl

perl is hard linked, therefore when parrot is removed,
the inode counter goes down and that is it.

iv) python3

nothing (bc of inode count being greater than 1)

D) Executing [2 marks]

```
ls -l progs/xcode-select
```

in `/tmp` gives:

```
-rwxr-xr-x 1 me me 118280 May 18 13:02 progs/xcode-select
```

For `me` to successfully execute:

```
progs/xcode-select
```

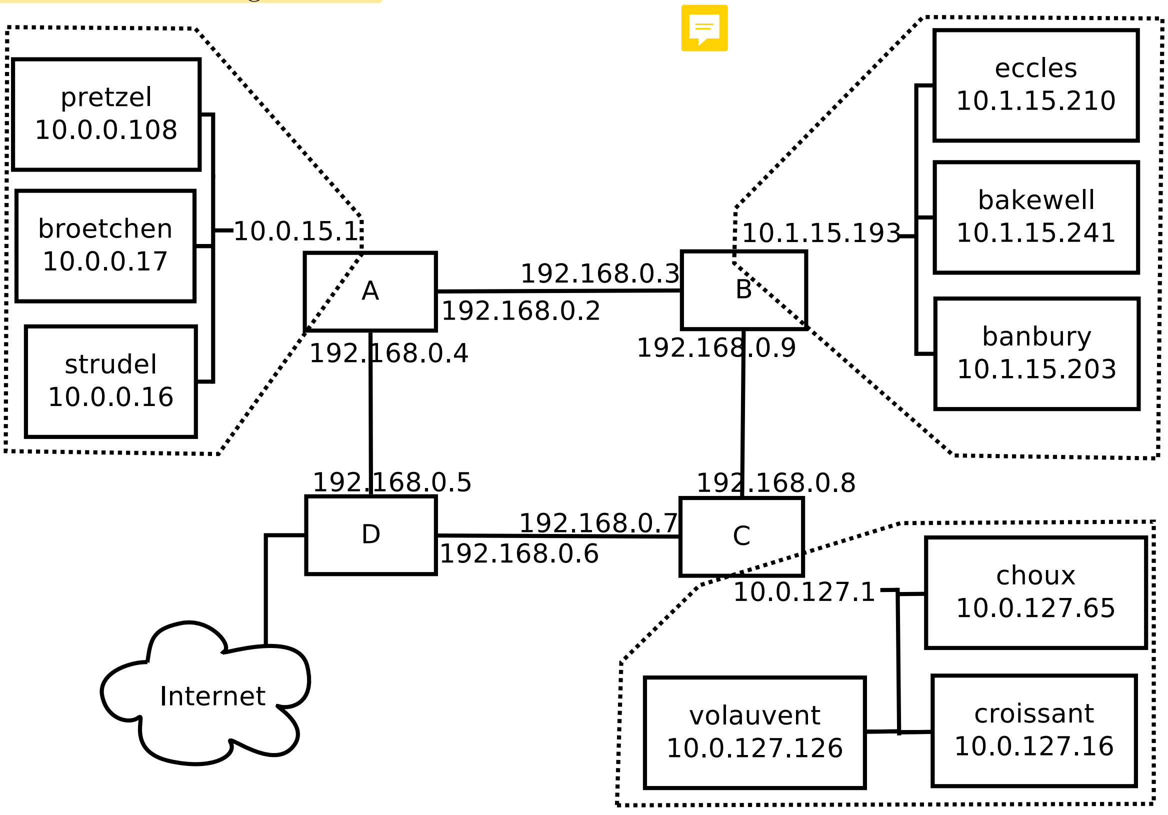
in the `/tmp` directory, what must be true about the `WINDOWS/SYSTEM32` directory? (Try to be as specific as possible).

../WINDOWS/SYSTEM32 must exist

the user 'me' is allowed to execute (change directory into) WINDOWS/SYSTEM32
and thus has the right to execute bits set on that directory

Question 6) [11 marks]

A) Consider the following network:



i) For the dotted network attached to each router, fill in the table. Give the broadcast address and netmask of each network as well as the number of unused interface addresses in the network (use the smallest possible networks): [5 marks]

Network	Broadcast address	Netmask	Unused addresses
A	10.0.0.127	255.255.255.128	127-2-3 = 122
B	10.1.15.255	255.255.255.192	63 - 2 - 3 = 58
C	10.0.127.127	255.255.255.128	127-2-3 = 122

2 for network and boardcast

3 for three used

The number of available hosts can be calculated using 2number of 0's in in subnet mask - 2 (minus 2 because of network and broadcast addresses)

ii) Which part of the network stack determines which path messages take from `strudel` to `choux`. [1 mark]

layer 3: network - MAC

iii) Which part of the network stack deals with "reliable" delivery? [1 mark]

layer4: transport - TCP

iv) Which part of the network stack would need its addresses modified in order for `bakewell` to communicate with the internet? [1 mark]

layer2: link

v) In order to function as a server, (a minimum) of four calls are required. Choose the four calls from the list and place them in the correct order: ssh, listen, select, bind, fdopen, socket, accept, htons, connect, system [3 marks]

socket,

bind,

listen,

accpet

Question 7) Consider a "unix" filesystem where: [8 marks]

- All i-nodes are cached in RAM
- i-nodes have 11-direct pointers, 1 indirect pointer and 2 double indirect pointers.
- Blocks are 16KB
- Block pointers are 16Bytes
- blocks are numbered from 0.

A) What is the maximum possible file size for this file system (in KB)? [2 marks]

Pointers per block: 16*1024 / 16 = 1024
Total no. of pointers: (11) + (1024)*(1) + (1024^2)*(2) = 2098187
Size = no. of pointers * block size = 2098187 * 16*1024 = 34376695808 bytes or
34376695808/ 2^10 = 33570992 KiB
ALT: 16 KiB * [(dir) + (indir)*1024 + (double_in)*1024^2] = 34376695808 KiB

B) How many blocks (in total) must be accessed to read the following blocks from a file (only count each access once):
$0, 10, 2058, 1049611$ [2 marks]

0: 1 for top level inode + 1 for block 0
10: 1 for block10 (inode is cached)
2058: 1 for double indirect + 1 for single indirect off that double + 1 for block 2058
1049611: 1 for single indirect off that double + 1 for block 1049611
total = 8

C) If the block size on this filesystem was 8KB instead of 16KB (all other parameters are held constant), what would the maximum possible filesize on this filesystem? [2 marks]

8 KiB * [(11) + (1)*512 + (2)*512^2] = 4198488 KiB

D) Give a benefit for using the smaller block size. [2 marks]

less internal fregmentation

Question 8) [9 marks]

A) Consider the following code which program A executes:

```
void f(void) {
    fork();
}

int main(int argc, char** argv) {
    pid_t me=getpid();
    int stat;
    fork();
    for (int i=0;i<3;++i) {
        f();
    }
    wait(&stat);
    exit(0);
}
```

i) How many processes are **created** (directly or indirectly) as a result of A's execution. [2 marks]

2^4 - 1= 15

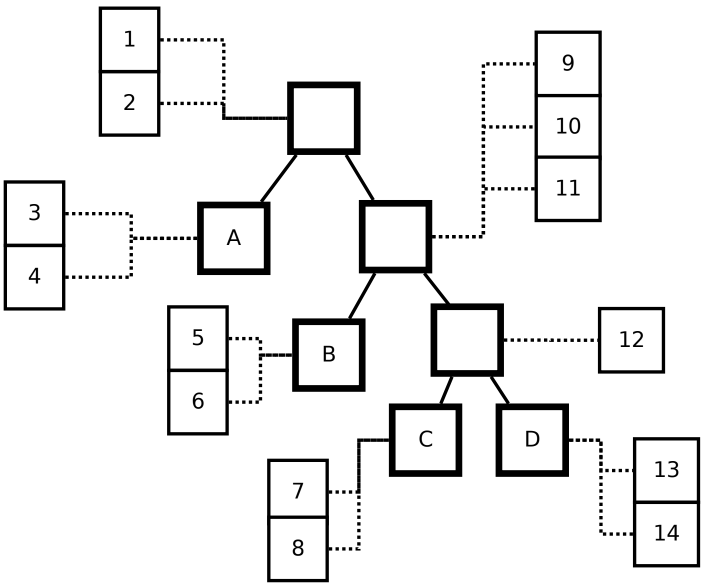
ii) How many zombies (created by the above code) are not reaped by some instance of A. [1 mark]

15 - 8 = 7
子进程会结束
但八个主线程child只会少一个

iii) Which system call is used to send signals to other processes? [1 mark]

kill(pid_t pid, int signum);

B)



In the diagram above, the heavy boxes represent processes. The numbered boxes represent commands to be run by those processes. The aim is to create the pipeline `A|B|C|D`. Choose functions from the list to fill in the boxes (some commands may be used in multiple boxes and some may be unused):

exec(), fork(), pipe(), dup2(), socket(), listen(), unlink() [5 marks]

1.	pipe()	5.	dup2()	10.	pipe()
2.	fork()	6.	exec()	11.	fork()
3.	dup2()	7.	dup2()	12.	fork()
4.	exec()	8.	exec()	13.	dup2()
		9.	dup2()	14.	exec()

Question 9) In all of the following you may omit `#include`s. You may assume that all lines in files have 79 or fewer characters. You may assume that all integers are positive (with no leading 0's). You do not need to consider arithmetic overflow. [18 marks]

A) Write a function `int sumline(const char* line)` which adds up the space separated integers in a string. `sumline("12 2 3 5")` would return 22. The function should process as much of the string as it can, but it will not move past a character which is not a space or part of a base 10 integer.

[5 marks]

```
int sumline(const char* line) {
```

B) Write a function `void sumfile(FILE* in, FILE* out)` which sums the integers on each line in `in` and outputs each sum to a new line of `out`. You may call the sumline function from the previous part and assume that it works correctly. [5 marks]

```
void sumfile(FILE* in, FILE* out) {
```

C) Write a program which takes one or more filenames as commandline parameters. For each filename `f`, create an output file `f.out` and sums up the lines in `f` and stores the results in `f.out`. You may call the `sumfile()` function from the previous part and assume that it works correctly.

Your program should use a separate process to deal with each input file. If any input/output pair can not be opened, continue processing the other pairs. [8 marks]

Question 10) In all of the following you may omit `#includes`. You may assume that all system calls succeed. You do not need to consider integer overflow. You may write additional functions to call.

Implement a function `int countsubstr(const char* big, const char* srch, int tcount)` which counts the number of times a string `srch` occurs in a string `big`. Eg:

```
countsubstr("bigsbigbigs", "big",1)
```

would return 3. The final paramter, determines the total number of threads to use. Your implementation does not need to be fast/efficient but it must make effective use of of threads. It must also be thread safe. [14 marks]

```
int countsubstr(const char* big, const char* srch, int tcount) {
```

END OF EXAMINATION

Working space

This page left blank

Working space

This page left blank