**CSSE2310: 2013 exam (2) answers**

**The exam was leaked in 2013 so the paper was rewritten last minute; this is for the real (actual) 2013 exam.**

[**UQAttic**](http://uqattic.net)

**Get more out of your study time. Join UQAttic's revision chat.**

[**Other exam papers**](https://drive.google.com/drive/folders/1k04K5RHL61k3ae2jJ362MPdN5xy3Bqsp)

Please **contribute** to these documents.

If you're looking for an effective way to familiarise yourself with the course material, you can't go past collaborating with fellow students. We have laboured to put these up, and so at the very least point out where you think we are wrong!

You'll get more out of the course, you'll do better in the exam, and other students will benefit from your input as well.

To get editing permissions, simply go to the [chatroom](http://uqattic.net) and provide us with your Google Account address.

**Style.**

Type answers in blue beneath each question.

If you're unsure of your answer, highlight your answer text then hit Ctrl+Alt+M to create a comment beside the text. Once you're satisfied with the answer, click the "Resolve" button on the comment.

If you want some extra explanation from someone else on their answer, highlight the other person's answer and repeat the procedure above.

**Communicate.**

Head over to [uqattic.net](http://uqattic.net/) and click "Chat Now!". You'll find a chatroom full of students just like you. Talk about a revision document (like this one) or swap prep tips. If you have your own IRC client, point it to irc.uqattic.net, port 6667, channel #attic.

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

A) Create a directory called things and move all .h files in the current directory into it.

mkdir things

mv \*.h things

mv \*\.h things (don’t forget to escape periods, they are regex for any character)

B) Show all lines in the file f1 which end with ;

grep \;$ f1

C) A file f2 consists of an unknown number of space separated columns. Display all columns execpt

the first two (from all lines in the file).

cut -d’ ‘ -f3- f2

cut -d’ ‘ -c3- f2 (this would also work)

D) Display the i-number of the file stuff.c

ls –i stuff.c

E) For files f1, f2, f3 show all lines from all the files which do not contain any of the words “song”,

“river” and “awful”.

grep –v “song” f1 f2 f3 | grep –v “river” | grep –v “awful”

cat f1,f2,f3 | grep -v “song” | grep –v “river” | grep –v “awful”

grep -v "song\|river\|awful" f1 f2 f3

cat f1 f2 f2 | grep -v -e song -e river -e awful

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

A) an array of 15 non-negative integers.

unsigned int foo[15];

B) a struct containing an array of 2 integers called fd, a character called x and a ﬂoating point value

called v.

struct {

int fd[2];

char x;

float v;

} foo;

C) a pointer to a function which takes two integer pointers and returns a character.

char (\*foo)(int\*,int\*)

D) a pointer to a non-modiﬁable integer.

const int \*foo;

int const \*foo; // also valid

E) another name for the boolean type.

typedef bool foo;

Q3)

A)

int a=5;

int b=12;

printf("%d", b^a);

9

B)

int a=5;

int b=12;

printf("%d %d", ++a, b&17);

6 0

C)

int a=5;

int b=12;

printf("%d %d", a|7, b||2);

7 1

Logical OR returns either a 1 or a 0. In this case b is TRUE (not 0) so 1 is returned.

D)

int a=5;

int b=12;

printf("%d", 2-a\*b);

-58

E)

int a=5;

int b=12;

int z=4;

for (int i=a;i<b;++i) {

int z=i+b;

}

printf("%d", z);

4

Note that “int z” is redeclared within the for loop, this is like a new version of z that only exists within that loop, any changes to it do not affect the original z. Since printf is outside the for loop it is referencing the original unchanged z.

F)

int a=5;

int b=12;

int c=a<<2;

int d=b>>2;

printf("%d %d", c, d);

20 3

G)

int a[]={2,3,4,5,6,7,8,9};

int\* b=a+a[0];

printf("%d", b[2]);

6

b is created pointing to the array a, then moved across by 2 (a[0]) positions, hence now points to the index 2 (which is the integer 4) in a. The print indexes b two more positions across from 4 landing on 6.

H)

char\* a="21";

int b=5+a[2];

printf("%d", b);

5

I)

for (int i=10;i<15;++i) {

switch (i%7) {

case 0:

case 1: printf("L"); break;

case 2: printf("LM"); break;

case 4: printf("H");

case 3: printf("M"); break;

default:

break;

}

}

MHML

J)

int a=5;

int b=12;

int c=(b%a>3)?1:0;

if (c)

printf("%d", b^4);

else

printf("%d", a^4);

1

K)

int f(int z) {

return (z%2)? ((z>0) ?(z/2)\*2 : z-1) : z;

}

...

printf("%d %d %d", f(5), f(-5), f(8));

4 -6 8

The return statement is two nested qualitative statements that evaluate to:

if(z%2 == 1){

if(z>0){

//don’t forget this is integer division

return ((z/2)\*2);

} else {

return z-1;

}

} else {

return z;

}

Question 4) A system has 32bit virtual addresses, 4KB pages and page table entries are 4Bytes. It uses a two level page table. [6 marks (2 each)]

A) Give the page number and offset for the following (base-10) addresses:

101010, 69632, 90210, 20479

101010/4096 = 24, offset 2706

69632/4096 = 17, offset 0

90210/4096 = 22, offset 98

20479/4096 = 4, offset 4095

B) What is the purpose of the TLB?

(Translation Lookaside Buffer)

To speed up page table lookup

C) Why does accessing the NULL pointer cause a segmentation fault?

A NULL pointer points to memory that doesn’t exist

Question 5) [17 marks]

A) What affect does fragmentation have on an indexed filesystem? [2 marks]

Memory is wasted. If a file is deleted and memory is freed, new memory may be too large to fit into fragmented sections.

B) When is a file removed from a “unix” file system? [2 marks]

A file is removed when the link count to the inode becomes 0.

C) Consider a “unix” filesystem where:

• i-nodes have 10-direct pointers, 1 indirect pointer and 1 double indirect pointer.

• Blocks are 8KB

• Block pointers are 4Bytes

• blocks are numbered from 0.

How many blocks are required to store files with the following sizes: [2 marks]

• 90107 bytes

ceiling(90107/8192) = 11 + 1 block for i-node + 1 indirect = 13 block

• 122897 bytes

ceiling(122897/8192) = 16 + 1 block for i-node + 1 indirect = 18 blocks

• 16793600 bytes.

ceiling(16793600/8192) = 2050 + 1 block for i-node + 1 indirect = 2052 (indirection block since 2050 > 10 and no double indirection since 2050 < 2048+10)

Consider the following directory listing:

4956250 drwxr-xr-x 4 susan extra 4096 Oct 29 12:12 .

4956264 drwxr-xr-x 3 susan extra 4096 Oct 29 11:47 ..

4956268 -rwxr-xr-x 1 ronald nobles 11616 Oct 29 11:48 2310port

4956284 lrwxrwxrwx 1 ronald nobles 8 Oct 29 12:12 7231port -> 2310port

4956262 -rw----r-- 1 fred watch 1657 Oct 29 11:46 cookbook.bib

4956263 -rw----r-- 1 fred watch 4255 Oct 29 11:46 cookbook.tex

4956256 -rw-r----- 1 sam nobles 139586 Oct 29 11:45 Data.cpp

4956257 -rw-r----- 1 sam watch 97905 Oct 29 11:45 Data.h

4956258 -rw-r----- 1 sam nobles 6696 Oct 29 11:45 DataVector.cpp

4956259 -rw-r----- 1 sam watch 6373 Oct 29 11:45 DataVector.h

4956260 -rw-r--r-- 1 nobby watch 871 Oct 29 11:45 \_\_init\_\_.py

4956261 -rw-r--r-- 1 nobby watch 67323 Oct 29 11:45 pdetools.py

4956270 drwxr-xr-x 5 susan extra 4096 Oct 29 11:55 src

4956274 -rwxr-xr-x 5 susan extra 4096 Oct 29 11:55 srcd

4956281 drwx--x--- 2 susan extra 4096 Oct 29 12:01 things

4956269 -rwxr-xr-x 1 sybil nobles 7472 Oct 29 11:48 tlimit

4956283 -rwx---r-x 1 ronald nobles 680 Oct 29 12:04 tscript.sh

4956254 -rw-r--r-- 1 sam nobles 6772 Oct 29 11:45 WrappedArray.c

4956255 -rw-r--r-- 1 sam watch 2598 Oct 29 11:45 WrappedArray.h

Group Members

watch - fred, nobby, sam

nobles - ronald, sam, sybil

extra - susan

D) What can nobby do to the following: [2 marks]

• pdetools.py

• 7231port

Read and write to pdetools.py. Not sure about the other as 7231 port is symlinked to 2310port with different permissions.

You take the first lot of permissions don’t you? Even if they’re more ‘restrictive’ than the second ones?

Mitch: You ignore the permissions of the link, you go by the permissions of the linked file (i believe).

E) Which groups would a new user william need to be a member of in order to read all of the

following: 2310port, tscript.sh, pdetools.py [2 marks]

any group that isn’t **nobles**

F) What command(s) could sam type to permit susan to read Data.h but prevent nobby from doing

so? [2 marks]

chmod 604 Data.h

chmod g-r,o+r Data.h # uses the symbolic chmod syntax

G) What would change in the directory listing after susan executed rm -rf srcd [2 marks]

srcd would be removed, and no confirmation would be required (the -f flag).

H) Which user(s) could successfully run gcc WrappedArray.c and why? [2 marks]

Only susan.

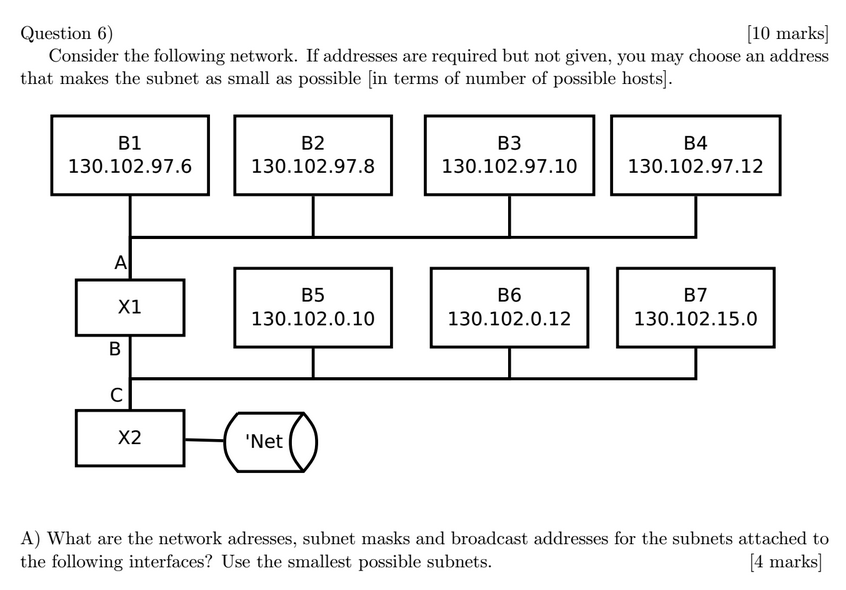
All users have read access to **WrappedArray.c** but gcc will create a derivative file within the directory. Only susan has write access to the directory as per

‘4956250 drwxr-xr-x 4 susan extra 4096 Oct 29 12:12 .’

I) How many subdirectories does things have? [1 mark]

0 Subdirectories,

because number of links to a directory is the number of subdirectories plus 2. More specifically when you create a new directory called **newDir,** there is the reference to **newDir** in the parent directory, and inside **newDir** is the **.** reference to itself. Thus 2 links for a newly made directory.



|  |  |  |  |
| --- | --- | --- | --- |
|  | Network Address | Subnet Mask | Broadcast Address |
| A on X1 | 130.102.97.0 | 255.255.255.240 | 130.102.97.15 |
| B on X1 | 130.102.0.0 | 255.255.240.0 | 130.102.15.255 |
| C on X2 | 130.102.0.1  \*We can chose address, so just pick an address for the C interface thats not already used | 255.255.240.0  \*C is on the same network as B, so same subnet | \*Not sure what todo here, just pick a different broadcast address?  Would it just be the same again?  So 130.102.15.255? or does it need to be different i.e. 130.102.15.254 |

B) Assuming no NAT is performed, what is the network for the organisation in /n form?[use the

smallest possible network]: [2 marks]

We take the network address of C and then find the number of bits, n that identify the network part of the address(the subnet)

subnet is 255.255.240.0 so 1111 1111 . 1111 1111 . 1111 0000 . 0000 0000

130.102.0.1/20

C) Without using larger subnets or NAT, how many additional machines could be connected to the

same subnet as Interface A? [1 mark]

Subnet on A = 1111 1111. 1111 1111 . 1111 1111 . 1111 0000 Only 4 bits that can change which means 16 addresses. The network address is 0, broadcast address is 15 and there is already 4 machines on the network. So only 10 more machines can be on the network.

**Side question:** to address the interface at A, do you use the network address? eg. when machine B1 creates the IP packet, it puts the address: 130.102.97.0 in it?

Does the network address not refer to the local address from the router and the broadcast address is what is sent to the larger network? So from inside A you’d use the network address and outside you’d use the broadcast address to interact with it?

Don’t think so, broadcast address is a special “send to everyone” address that you’d only use in particular situations. I don’t think it is used for regular traffic.

D) To which layers do the following belong: [3 marks]

|  |  |
| --- | --- |
| Term | Layer |
| TCP | Transport |
| IP | Network |
| UDP | Transport |
| HTTP | Application |
| SSH | Application |
| Ethernet | Link |

Question 7) [7 marks]

A) What is the c type of the exit value of a thread in the pthreads API? [1 mark]

void\*

B) Which function can be called on a thread so it doesn’t need to be “joined”? [1 mark]

pthread detach

C) Which single function call will end all threads in a process? [1 mark]

exit()

D) Suppose a child process with process id pid has just finished. The following code executes:

waitpid(pid, &s, 0);

What value did main in the child return? [1 mark]

WEXITSTATUS to get

information about exit status

E) A process with pid = 18817 appears in the process list.

kill 18817

is run from the shell, but the process is still in the process list.

Give two possible reasons for this. [3 marks]

18817 is ignoring the kill command, kill sends a SIGTERM signal(SIGTERM requests termination) by default which can be caught by the process and ignored.

or 18817 is a zombie process.

Also another possibility from the newsgroup: The user executing kill 18817 does not have access to the kill command for that process

Another possibility:

* has been killed, but not yet repeated by init (defuc).

* init will not adopt if parent is still alive. So, 18817 could be a child process, that has been killed.

sigaction(SIGTERM, SIG\_IGN, NULL); // we can ignore the signal

sigaction(SIGTERM, handler, NULL); // we can have a handler that just doesn’t do anything

void handler(int sig) { return; }

Question 8)

In all of the following you may omit #includes. You may assume that all system calls succeed and that all processes exit normally. You may assume that all lines in files have 79 or fewer characters. Where tasks run in parallel, you do not need to worry about the interleving of output. [34 marks overall]

A) Implement a function substLines which takes two strings and a filename. It should print each

line in the file with characters in the first string replaced by the corresponding character in the second

string. Each line should be preceeded by two dots. If the second string is shorter than the first

string, any extra characters in the first string should be ignored. It should return true if the file could

be opened successfully and false if not. [6 marks]

For example, under substLines("ao" ,"ua", filename)

Twas brillig and the slithy toves ⇒ ..Twus brillig und the slithy taves

did gyre and gimble in the wabe ⇒ ..did gyre und gimble in the wube

**bool substLines(const char\* s1, const char\* s2,const char\* filename) {**

FILE \*file;

char line[81];

if ((file = fopen(filename, "r")) == NULL) {

return false;

}

while(fgets(line, 81, file) != NULL) {

for(int i = 0; i < strlen(line); ++i) {

for(int j = 0; j < strlen(s2); ++j) {

if(line[i] == s1[j]) {

line[i] = s2[j];

break;

}

}

}

printf(“..%s”, line);

}

return true;

}

// The above solution has a bug in it. If you do not break; after performing the replacement ( line[i] = s2[j]; ) then you may end up doing erroneous replacements.

Assume we have a file **file1.txt** with a single line of content:

aaaa

// If we call the above implementation we will get the wrong output

**substLines(“ab”, “bc”, “file1.txt);**

**..**cccc

This is because we perform the replacement of ‘a’ with ‘b’, and then keep checking if the current character in **line** matches any other characters in **s1**. The character in line has changed to ‘b’ now, so it will match the next char in **s2** as well, triggering another replacement.

B) Implement a function trSubst which takes a two strings and a filename. It should print each line in

the file with characters in the first string replaced by the corresponding character in the second string.

Each line should be preceeded by two dots. It will do this by invoking the tr program (which will be

on the path). The tr program takes two string parameters and reads the data to be tranformed from

stdin and sends its output to stdout. It should return true if the file could be opened successfully

and false if not. You may not use the functions system(), or popen(). [6 marks]

**bool trSubst(const char\* s1, const char\* s2, const char\* filename) {**

#define READ 0

#define WRITE 1

int fdpc[2];

int fdcp[2];

pipe(fdpc);

pipe(fdcp);

pid\_t pid;

FILE\* file;

char sendLine[81];

char recvLine[81];

if ((file = fopen(filename, "r")) == NULL) {

return false;

}

if (fork()) { /\* Parent \*/

close(fdcp[WRITE]);

close(fdpc[READ]);

FILE\* toTr = fdopen(fdpc[WRITE], "w");

FILE\* fromTr = fdopen(fdcp[READ], "r");

while(fgets(sendLine, 81, file) != NULL) {

fprintf(toTr, “%s”, sendLine);

fflush(toTr);

fgets(recvLine, 81, fromTr);

printf(“..%s”, recvLine);

}

} else { /\* Child \*/

close(fdcp[READ]);

close(fdpc[WRITE]);

dup2(fdcp[WRITE], 1);

dup2(fdpc[READ], 0);

execlp("tr", “tr”, s1, s2, NULL);

exit(0);

}

return true;

}

bool trSubst(char\* s1, char\* s2, char\* filename) {

const int R = 0;

const int W = 1;

FILE\* file = fopen(filename, "r");

if (!file) return false;

int pcFds[2]; // parent-child pipe

int cpFds[2]; // child-parent pipe

pipe(pcFds);

pipe(cpFds);

pid\_t pid = fork();

if (pid == 0) { // child

close(pcFds[W]);

close(cpFds[R]);

dup2(pcFds[R], STDIN\_FILENO);

dup2(cpFds[W], STDOUT\_FILENO);

execlp("tr", "tr", s1, s2, 0);

}

else { // parent

close(pcFds[R]);

close(cpFds[W]);

FILE\* pc = fdopen(pcFds[W], "w");

FILE\* cp = fdopen(cpFds[R], "r");

char buf[80];

while (fgets(buf, 80, file) != NULL) {

fprintf(pc, "%s", buf);

fflush(pc);

}

// tr output will be fully-buffered. We have to close the pipe

// so that tr hits EOF. When this happens it flushes the buffer and closes.

fclose(pc);

while (fgets(buf, 80, cp) != NULL) {

printf("..%s", buf);

}

fclose(cp);

wait(NULL);

}

return true;

}

C) Implement a function substLinesMany which takes two strings and an array of filenames (and its

size) and calls the function from Part A on each filename with the two strings. (You may assume the

function exists and works correctly). All the specified files should be processed concurrently. If any of

the files result in an error, then substLinesMany will return false otherwise return true. [8 marks]

**bool substLinesMany(const char\* s1, const char\* s2, const char\*\* filenames, int numfiles) {**

int i = 0;

int status;

bool result = true;

for(i = 0; i < numfiles; i++) {

if (fork() == 0) {/\* Child \*/

if (substLines(s1, s2, filenames[i])) {

exit(0);

} else {

exit(2);

}

}

}

for(i = 0; i < numfiles; i++) {

wait(&status);

if (WIFEXITED(status)) {

if (WEXITSTATUS(status) == 2) {

result = false;

}

}

}

return result;

}

**// Question 8C using threads**

typedef struct Arg {

const char\* s1;

const char\* s2;

const char\* file;

bool\* success;

} Arg;

void\* handler(void\* arg) {

Arg\* args = (Arg\*) arg;

if (!substLines(args->s1, args->s2, args->file)) {

\*(args->success) = false;

}

return NULL;

}

bool substLinesMany(const char\* s1, const char\* s2,

const char\*\* filenames, int numfiles) {

pthread\_t\* tids = malloc(sizeof(pthread\_t) \* numfiles);

bool\* success = malloc(sizeof(int));

\*success = true;

for (int i = 0; i < numfiles; i++) {

Arg\* args = malloc(sizeof(Arg));

args->s1 = s1;

args->s2 = s2;

args->file = filenames[i];

args->success = success;

pthread\_t tid;

pthread\_create(&tid, NULL, handler, (void\*)(args));

tids[i] = tid;

}

for (int i = 0; i < numfiles; i++) {

pthread\_join(tids[i], NULL);

}

return \*success;

}

Does anyone know how to do these ones properly?

Apparently C was intended for threading and D requires forking

D) Implement a function trSubstMany which takes two strings and an array of filenames (and its

size) and calls the function from Part B on each filename with the two strings. (You may assume the

function exists and works correctly). All the specified files should be processed concurrently. If any

of the files result in an error, then trSubstMany will return false otherwise return true. [8 marks]

**bool trSubstMany(const char\* s1, const char\* s2, const char\*\* filenames, int numfiles) {**

int i = 0;

int status;

bool result = true;

for(i = 0; i < numfiles; i++) {

if (fork() == 0) {/\* Child \*/

if (trSubst(s1, s2, filenames[i])) {

exit(0);

} else {

exit(2);

}

}

}

for(i = 0; i < numfiles; i++) {

wait(&status);

if (WIFEXITED(status)) {

if (WEXITSTATUS(status) == 2) {

result = false;

}

}

}

return result;

}

E) Write a program which takes the following arguments: [6 marks]

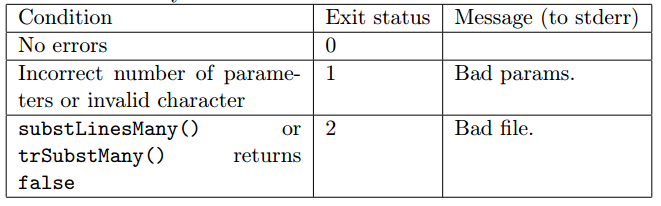
• two strings

• a sequence of filenames to process.

For each file name, it should perform the substitute operation using substLinesMany() and then

trSubstMany(). For this part, you may assume that substLinesMany() and trSubstMany() exist

and function correctly.



**int main(int argc, char\*\* argv)**

**if( argc < 3) {**

**fprintf(stderr, “Bad params.”);**

**exit(1);**

**}**

if(substLinesMany(argv[1], argv[2], argv[3]) == 0 ) {

**fprintf(stderr, “Bad file.”);**

**exit(2);**

}

if(trSubstMany(argv[1], argv[2], argv[3]) == 0 ) {

**fprintf(stderr, “Bad file.”);**

**exit(2);**

}

exit(0);

}

This is wrong ^   
A sequence of files is in the form ./a.out “abc” “xyz” f1 [f2 …]

**int main(int argc, char\*\* argv)**

**if( argc < 4) {**

**fprintf(stderr, “Bad params.”);**

**exit(1);**

**}**

for(int i =3; i < argc; i++) {

if(substLinesMany(argv[1], argv[2], argv[i]) == 0 ) {

**fprintf(stderr, “Bad file.”);**

**exit(2);**

}

if(trSubstMany(argv[1], argv[2], argv[i]) == 0 ) {

**fprintf(stderr, “Bad file.”);**

**exit(2);**

}

}

exit(0);

}