**CSSE2310: 2021 SEM2 exam answers**

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

[**Other exam papers**](https://drive.google.com/drive/folders/1i3euI6FbNv_u3fOmAtlMRzZrSUb5vYXf?usp=sharing)

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**, visit [UQAttic.net](https://uqattic.net) and provide us with your Google Account address by clicking on the “Get editing permissions” button on the front page.

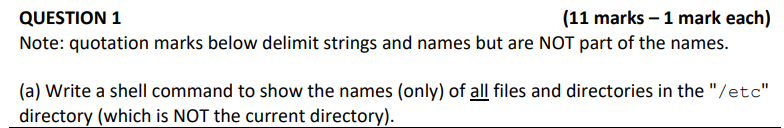
**Style.**

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.

It’s the day of the CSSE2310 2022 Sem 1 exam so time to try the one from last sem lmao



a) ls -a ~/etc

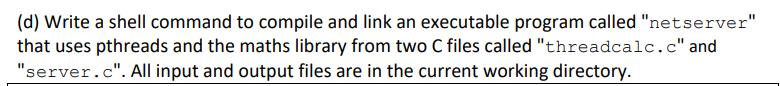
##should this be “ls -a /etc” ? Yes, “~” is $HOME. [+4]

Mv /tmp/a4data/\*.spec ~/data [+1]



Mkdir exam

All commands should start with lowercase letters i.e mkdir exam



Gcc -lpthread -lm threadcalc.c server.c -o netserver (unsure)

Gcc -pthread -lm threadcalc.c server.c -o netserver (to link pthread it’s just -pthread, no “l”)

Does this work aswell?: gcc -pthread -lm **-c** threadcalc.c server.c -o netserver [+1]



svn commit

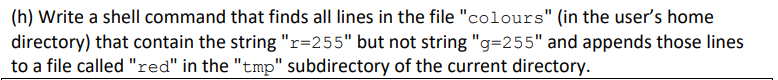


Grep tcp /etc/services

grep “tcp” /etc/services



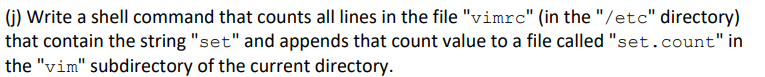
Grep -v nameserver /etc/resolv.conf



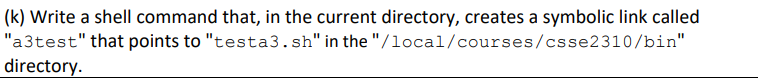
Grep r=255 ~/colours | grep -v g=255 >> ./tmp/red (why the . in front of /tmp?)`I think because it is subdirectory of current directory [+1] Doing temp/red will also work



Grep -c “Mount Cootha” addresses



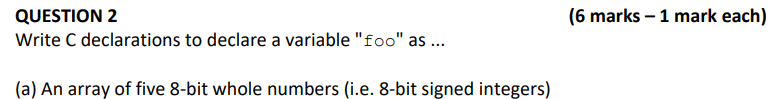
Grep -c set /etc/vimrc >> ./vim/set.count



Ln -s /local/courses/csse2310/bin/testa3.sh a3test

Does this have to be a soft link or could it be without the “-s”?

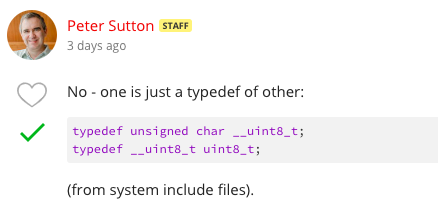
It is a symbolic link, so you need the -s [+1]



Uint8\_t foo[5]; (-1)

int8\_t foo[5]; -> asks for signed integers [+2]

char foo[5]; [+1]





void (\*foo)(int);



Pthread\_t foo; [+1]

pid\_t foo;



Float foo = 3.14;

float foo = 3.14f;



Int foo[2]; [+1]



Struct {

Char c;

Int i;

} foo; [+1]

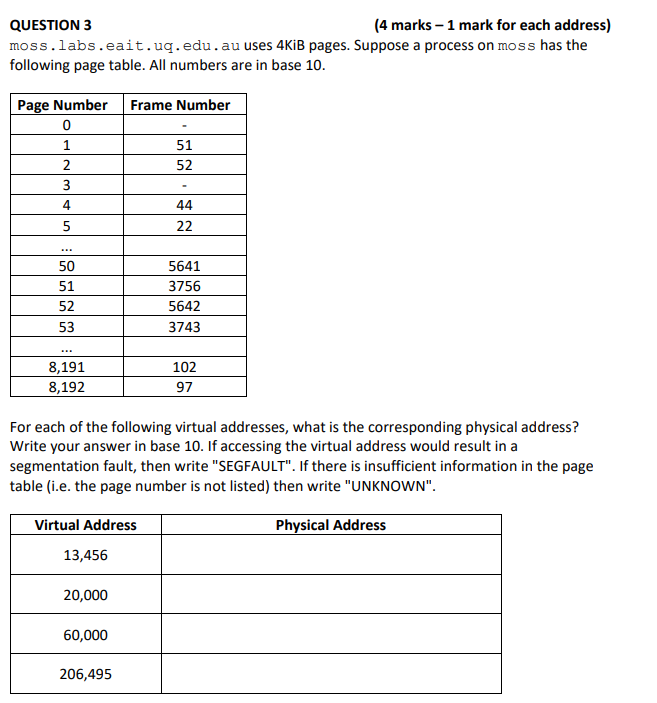
Struct Thing {

Char c;

Int i;

};

Struct Thing foo;

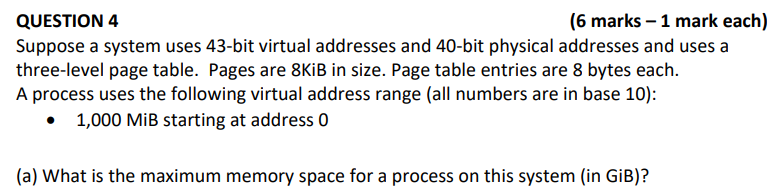


13,456 - SEGFAULT [+3]

20,000 - 183,840 [+3]

60,000 - UNKNOWN [+3]

206,495 - 23,107,231 [+3]



2^43 Bytes = 2^13 GiB



Entries per page = 8KiB/8 = 1KiB = 1024 Bytes

3rd level size = 1024 \* 8KiB = 8192KiB = 8MiB

2nd level size = 1024 \* 8MiB = 8192MiB = 8GiB

Num 3rd level page = 1000MiB/8MiB = 125 pages

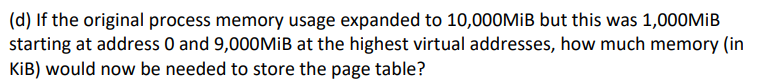
Total mem = (1 + 1 + 125) \* 8KiB = 1016KiB



Num 3rd level page = 10,000MiB/8MiB = 1250 pages

Now need at least 2 2nd level pages to store references

Total mem = (1 + 2 + 1250) \* 8KiB = 10024KiB



From b), we’re using (1 + 1 + 125) \* 8KiB for the 1,000MiB starting at address 0.

For the 9,000MiB:

Number of pages = 9000MiB / page size = 1,152,000 pages

1,152,000 / 1024 = 1,125 level 3 pages

1,125 / 1,024 = 2 level 2 pages

1 level 1 page

So in total, mem = ((1 + (1 + 125) + (2 + 1125) \* 8KiB = 1254 \* 8KiB = 10,032 KiB

1 page lvl 1 then 1 page lvl 2 to get to 125 pages at lvl 3 then 2 page lvl 2 to get to 1225 pages lvl 3

Explanation pls? 2022 Exam consultation 4 1:15:55

GUYS IM GONNA FAILLLLLLLLLLLLL - nah you got this bro -❤️thanks -

Question: Guys why is john williams so knowledgeable.

Answer: I think this makes sense, since u only need to access level 1 once right?

Top Level Table:

2^20 entries \* 8 bytes per entry = 2^23 bytes = 8 MiB = 8192 KiB

8192KiB + 125 \* 8KiB

= 9192 KiB (+4)

# should be 8(1 + 125) KiB

2^30 entries \* 8 bytes each

= 2^20 \* 2^10 \* 2^3 bytes

= 8192 MiB (+3)

# is it not 8 \* 125 KiB?

# 1000 \* 2^20 / 2^13 = 125 total pages

# 125 in layer 1

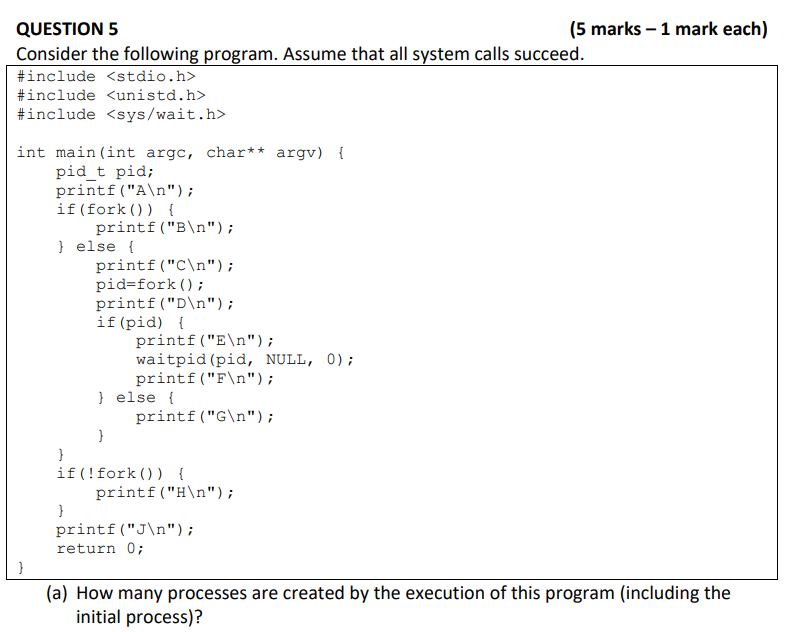
# 8 \* 125 KiB

Explanation please?

Exam consultation 2 covers a similar example but;

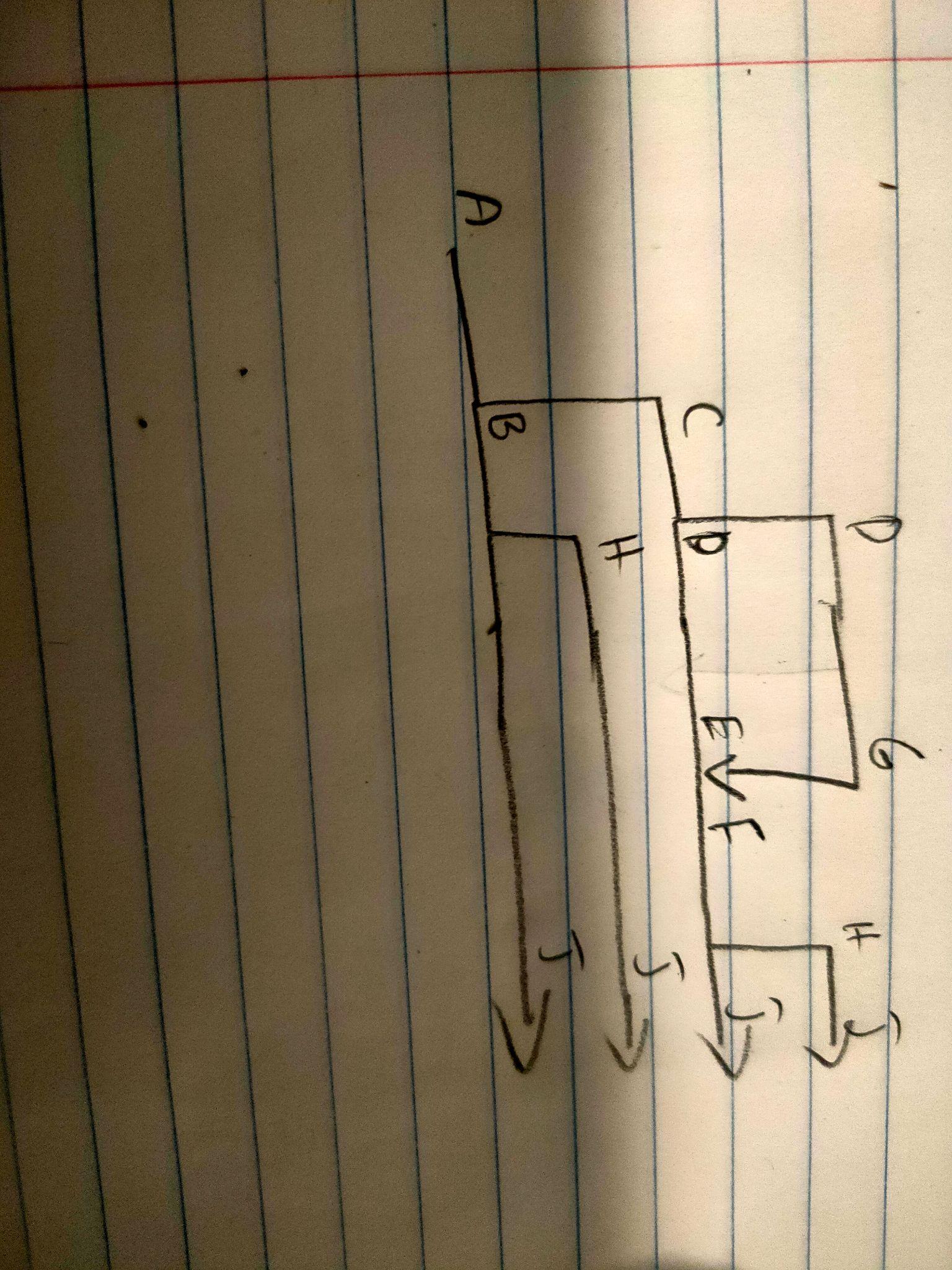
* For 9f
  + Consider the visual address, and how there are a total of 43 bits, with 13 being used for offset. If only a single level exists, that means that there exists only one block with a total of 30 bits assigned by the visual address to that one block.
  + So, the one first level block contains 2^30 entries
  + Each entry is 8 bytes, so 2^30 \* 8 = 2^33 bytes = 2^13 MiB = 8192MiB
* For 9e
  + From previous questions, we know that each level contains 10 bits for a 3-level visual address in this example, so it would have been   
    | 10 bits | 10 bits | 10 bits | 13 bits offset |  
    With the change from 3 levels to two, the remaining bits after the 2nd level become part of the first, so it becomes  
    | 20 bits | 10 bits | 13 bits offset |
  + The size of the first level page is equal to the number of entries \* entry size  
    = 2^20 \* 8  
    And the size of the 2nd level entries are 8KiB each
  + We also know the number of second level entries from before, which was 125 (just do address range (1,000 MiB) / page size (8 KiB)). Find size of second level with num entries \* page size = 125 \* 8KiB

Why do we not consider it when we have 3 tables? It would only be 4096 bytes so is it so small that we just don't bother to include it?



6 (+4)

* I got 5 (nevermind, got 6 now…below diagram is incorrect at point G)
* I believe he means that at point G you should create a child process, since one of the last lines (!fork()) causes all processes to fork, however only the child processes print “H”.





5 (+4)



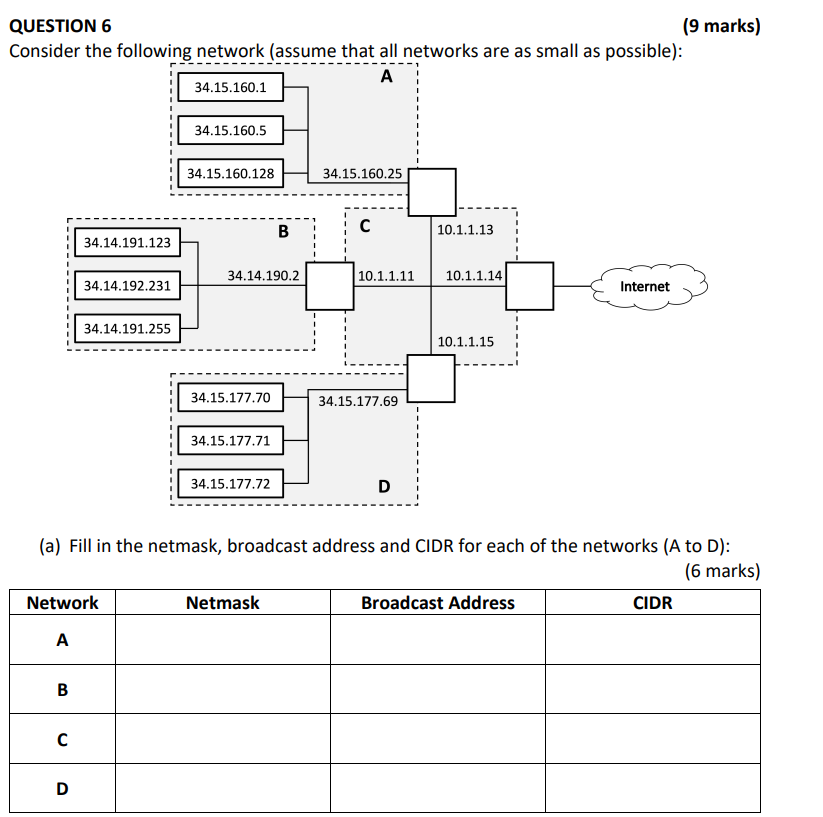
4 (+4)



17 (+4)



31 (+4)



Network A:

Lowest address - 34.15.160.1 = 34.15.160.0000 0001

Highest address - 34.15.160.128 = 34.15.160.1000 0000

CIDR = 34.15.160.0 / 24

Netmask = 255.255.255.0

Broadcast = 34.15.160.255

Network B: (Isnt this Network A? +1)

Lowest address - 34.15.160.1 = 34.15.160.0000 0001

Highest address - 34.15.160.128 = 34.15.160.1000 0000

Network B:  
Mask: 255.255.128.128 ]

Broadcast: 34.14.190.127

CIDR: 34.14.190.0/25

Mask: 255.255.128.0 [+2]

Broadcast: 34.14.255.255

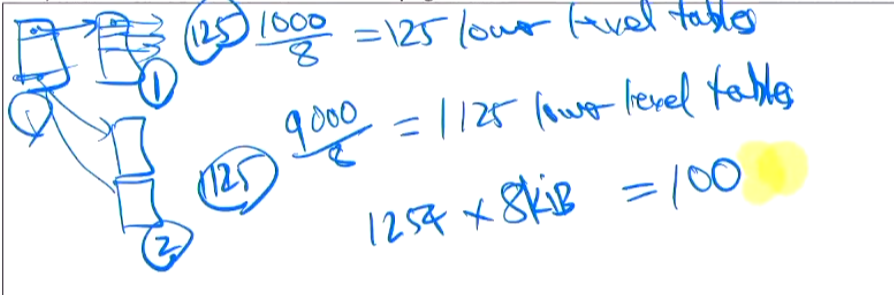
CIDR: 34.14.128.0/17 (+5)

SOMEONE EXPLAIN ME HOW WE GOT THESE ANSWERS FOR NETWORK B PLS

Network C:

Mask: 255.255.255.224

Broadcast: 10.1.1.31

CIDR: 10.1.1.0/27  
#(Can someone please explain how /27 was reached? I got /29 for network C)

We have to make sure that 10.1.1.15 is not the broadcast address. So the smallest range of low bytes permissible is 0-31, which means there are 32-5=27 network bits (+1)

Network D:

Mask: 255.255.255.240

Broadcast: 34.15.177.79

CIDR: 34.15.177.64/28

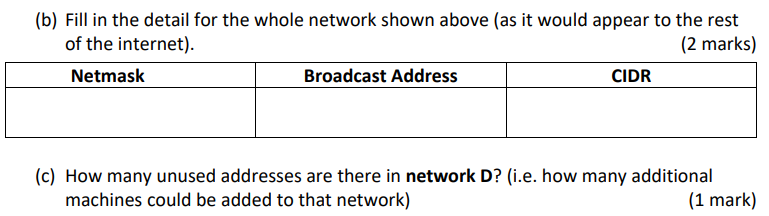
Whole network (what numbers were picked here?)

Mask: 255.254.0.0

Broadcast: 34.15.255.255

CIDR: 34.14.0.0/15

+3



2^4 - 3 = 13 (CIDR is /28?, so 32 - 28 = 4 bits for ports. 2^4 = 16. 3 already in use by the network, so 16 - 3 = 13 total remaining addresses. Would be good to get some confirmation on this)[-1] what a nerd can’t even get the right answer lmao

\*look at CIDR dash/ value == 28

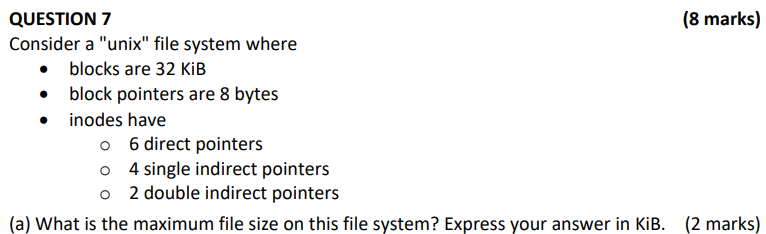
\*32 is 255.255.255.255 so 32-28 = 4

\*2^4=16

\*subtract address in use 4 addresses and 11111111 and 0000000

\*In total there are 10 addresses left.

/28 -> 16 addresses, 4 in use, 2 reserved, therefore 10 unused addresses. (+3)



(6 + (2^10 \* 4) + (2^20 \* 2)) \* 32KiB = 67,240,128 KiB[-1]

Thought it would be

32KiB/8 bytes = 4096 pointers per block

(6 + (4096 \* 4) + ((4096^2) \* 2)) \* 32KiB = 1,074,266,304 KiB[+8]



(6 + 4096 \*4) \* 32 = 524480 KiB [+3]



4 -> 3 for block reads at file, 1 for single indirect at block 6 [+3]

4 - don’t get the above explanation, but the memory starts in direct pointer 4, uses 5 and 6, and then uses the first pointer in the first single indirect, for a total of 4 blocks.

Above is Error

4 - in direct pointer 4, uses 5 and 6, and then uses 7 the first pointer in the first single indirect, for a total 6

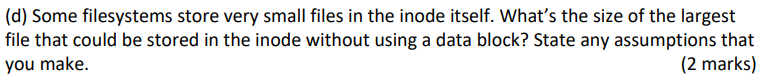
floor(200000/32768) = 6

Floor(150000/32768) = 4

Need to access blocks 4, 5 and 6

We know direct = 6 -> blocks 0 to 5

Block 6 is in single-indirect pointer node

3 + 1 = 4

Inode

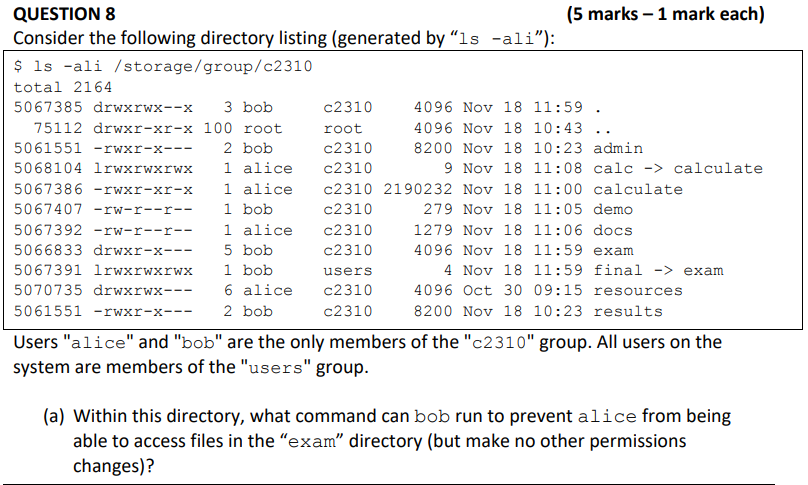
| Permissions |
| --- |
| Owner |
| Timestamp |
| Block pointers  (12 total for this question) |
| Meta Data |
| Size |

Can store data in block pointer section of inode such that, 12 pointers \* 8 bytes per pointer = 96 bytes total storage [+3]

Assumption: that there is a way to mark an iNode as containing binary data and not a pointer to a block, as otherwise the kernel may attempt to use binary data as a block address/index.

^^ This was a guess unsure if it’s right

Assumption: There aren’t other “fields” of the inode in which to store more data.



Chmod g-r exam (unsure)

Chmod g-rx (I think get access to file inside a dic means interacting with the dic, which needs x

(I tried on moss: without x, I cannot cd into the dic [+1]

“-r” bad (no other permissions changes)

Chmod g-x exam (to access file you need x. To see that the file exists in a directory, you need r so you can use ls). Having x with r, just means you can open an invisible file (file not listed when using ls). Try opening such invisible file on moss and you will see) [+1]



Chmod o+x resources (unsure)

Chmod o+r resource chmod g-r resource(tested on moss, [ls] is related to r

Error （./ have no other read drwxrwx--x 3 bob c2310 4096 Nov 18 11:59 .）??????

chmod g-r,o+r resources



4? 6 minus link between itself and “.”? [+3]

# 4, need to consider reference to itself (“.”) and the parent (“..”)



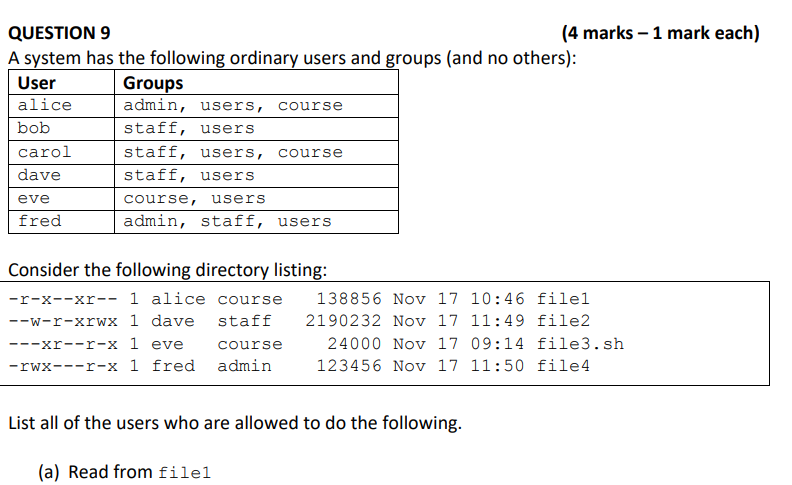
0, link still exists with file results[+2]

Hard-linked with admin (same inode number)



Alice, bob ->permissions are given to all with r, x for user, group and other, so i’d assume anyone can use this transaction [+2]

Note: Calc is symlinked to calculate



alice, bob, dave, eve, fred ? anyone not part of course as far as i can tell as well as alice (the user) all correct, except for eve, who is in course. [+5]



Alice, dave, eve fred all correct, except for fred, who is in staff [+5]



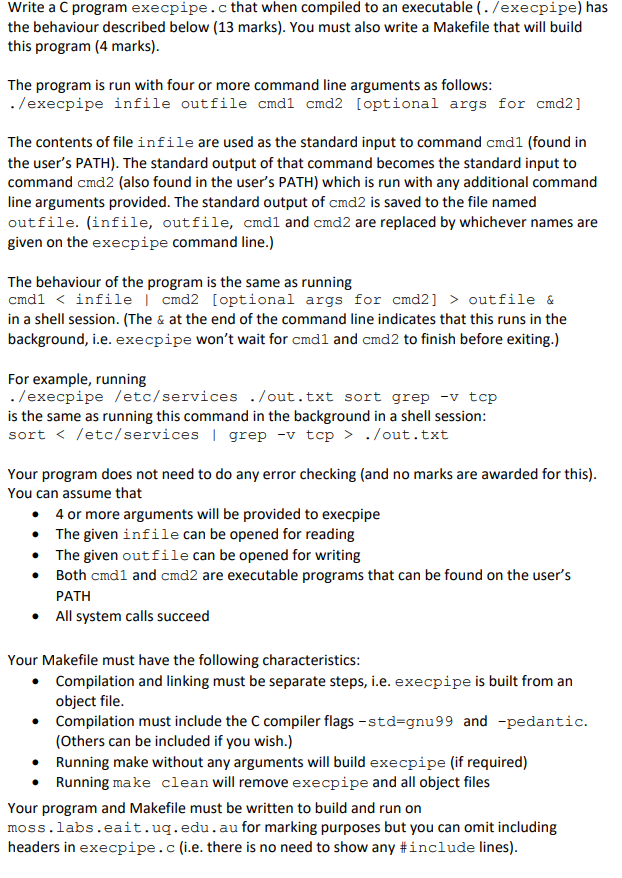
Bob, dave, fred, eve

Bob, dave, fred (you need both read AND execute for .sh scripts) [+5 ]



Bob, carol, dave, eve, [get help], fred [+4]

Q10)



1 #include <stdlib.h>

2 #include <stdio.h>

3 #include <unistd.h>

4 #include <sys/types.h>

5 #include <sys/wafcntl.hit.h>

6 #include <fcntl.h>

7

8 int main(int argc, char\* argv[]) {

9 int infile = open(argv[1], O\_RDONLY);

10 int fd[2];

11 pipe(fd);

12 dup2(fd[1], STDOUT\_FILENO);

13 dup2(infile, STDIN\_FILENO);

14 pid\_t pid;

15

16 if (pid = fork()) {

17 close(fd[1]);

18 } else {

19 // child

20 execlp(argv[3], argv[3], NULL);

21 }

22

23 int outfile = open(argv[2], O\_WRONLY);

24 dup2(fd[0], STDIN\_FILENO);

25 dup2(outfile, STDOUT\_FILENO);

26 execvp(argv[4], argv + 4);

27 return 0;

28 }

# This is what I got to work

1 #include <>

2 #include <stdio.h>

3 #include <stdlib.h>

4 #include <sys/wait.h>

5 #include <unistd.h>

6

7 int main(int argc, char \*argv[]) {

8 int infile = open(argv[1], O\_RDONLY), fd[2];

9 pipe(fd);

10 pid\_t pid;

11

12 if ((pid = fork())) {

13 close(fd[1]);

14 } else {

15 dup2(infile, STDIN\_FILENO);

16 dup2(fd[1], STDOUT\_FILENO);

17 close(fd[0]);

18 close(fd[1]);

19 execlp(argv[3], argv[3], NULL);

20 }

21

22 close(infile);

23 int outfile = open(argv[2], O\_WRONLY | O\_CREAT, 0777);

24 dup2(fd[0], STDIN\_FILENO);

25 close(fd[0]);

26 dup2(outfile, STDOUT\_FILENO);

27 close(outfile);

28 execvp(argv[4], argv + 4);

29 return 0;

30 }+

Seems to work but maybe not the best way…. I wonder if its ok to just exec /bin/bash lmao

MakeFile

CC=gcc

CFLAGS=-Wall -pedantic -std=gnu99

.PHONY:=clean

all: execpipe

#Generate executable by linking object files

execpipe: execpipe.c

$(CC) $(CFLAGS) execpipe.c -o execpipe

clean:

rm -f \*.o

rm -f execpipe

I wouldn’t call the above “compiling and linking in separate steps”. Better to use an additional “execpipe.o” target.