

note @367

381 views

## [Exams] Past Exams 2019 Q&A

Discuss all questions pertaining to exams which took place in 2019 here.

You can find the past exams here: <https://cs61c.org/resources/exams>

When posting questions, you **MUST** reference the semester, exam, **AND** question so we can help you. Please put this at the beginning of your post in this format: **[[Semester]-{Exam}]:Q{Question Number}**  
For example: **[SP-MT1]:Q1**, or **[SU-MT2]:Q3**

{Semester} is one of these: SP, SU, FA

{Exam} is one of these: Q, MT, MT1, MT2, F

If you follow this format, it will make it very easy to search for similar questions!

Updates:

[2/17 4:32PM] You can check out the video walkthroughs for sp19-mt1 made by our lovely tutor Evan Sum here: [https://www.youtube.com/watch?v=ZxMfbSXSJS0U&list=PLXdl-bKzbrT5qIMUdt3\\_M8FA764eIIWdG](https://www.youtube.com/watch?v=ZxMfbSXSJS0U&list=PLXdl-bKzbrT5qIMUdt3_M8FA764eIIWdG)

[5/9 6:54PM] Here's a video walkthrough for the Sp19 Final: <https://www.youtube.com/watch?v=8DiN5Hu9x24&list=PLDol-XvXOOapuEacxuUrUaBq2YDuKYptV&index=2> (handout and timestamps in comments)

[5/11 4:38 PM] Here's a video walkthrough for Su19 Final made by another one of our tutor's Sunay Poole: [https://www.youtube.com/watch?v=FY5dAMrXvxo&list=PL1jzNYdiQ6Dp4Wd23qGEGz0jAIPv-ZHyZ&index=5&t=21s&ab\\_channel=SunayPoole](https://www.youtube.com/watch?v=FY5dAMrXvxo&list=PL1jzNYdiQ6Dp4Wd23qGEGz0jAIPv-ZHyZ&index=5&t=21s&ab_channel=SunayPoole)

midterm\_exam1

midterm\_exam2

final\_exam

~ An instructor (Jie Qiu) thinks this is a good note ~

Updated 1 month ago by Stephan Kaminsky and 3 others

followup discussions *for lingering questions and comments*

1 endorsed followup comment

Resolved  Unresolved

This was marked a duplicate to the question/note above by Stephan Kaminsky 4 months ago

""



[REDACTED] 4 months ago

[SP-MT1]:Q1c

questions about past exams

```

Solution: void FreeAST (simple_AST *tree) {
    if (tree != NULL) {
        shared_string_t *lst = create_list ();
        FreeASTHelper (tree, lst);
        int i;
        for (i = 0; i < lst->size; i++) {
            free(lst->arr[i]);
        }
        free(lst->arr);
        free(lst);
    }
}

```

This is from sp19 MT1 1(c). I wonder, if we want to free an array. Do we have to free each single element and then free the array itself after just like this solutions shows? What if we just do free(lst->arr)?

I think it is said that arr pointer points to the first element of arr.

helpful! | 0



**Zac Patel** 4 months ago The answer to your questions depends on what type the array is. If the array is

```
int x[10] = malloc(sizeof(int) * 10);
```

then

```
free(x);
```

is sufficient. This is because the only memory that was allocated was the array itself.

If instead, I say malloc an array of integer pointers that are themselves malloc'd, then I must free the elements and then free the array.

ex:

```
int *x[10] = malloc(sizeof(int *) * 10);
x[0] = malloc(sizeof(int));
```

In this case, if we were to simply

```
free(x);
```

then we would end up losing the location to all the memory addresses we malloced, which would result in a memory leak.

---

Back to the problem - in this case lst->arr is an array of strings (char \*'s) each of which has been malloced, so we must free each element before freeing lst.

helpful! | 4



**[REDACTED]** 4 months ago Get it! Thanks!

helpful! | 0

Resolved  Unresolved

This was marked a duplicate to the question/note above by Stephan Kaminsky 4 months ago



**Anonymous Poet** 4 months ago

**[SP-MT1]:Q1a**

Where is the double free?

Don't we only free filename once, as we only call the freeing function once?

(a) Nick decides to use the following function to free his ASTs.

```
void FreeAST (simple_AST *tree) {
    if (tree != NULL) {
        int i;
        for (i = 0; i < tree->size; i++) {
            FreeAST (tree->children [ i ]);
        }
        free (tree->children);
        free (tree->filename);
        free (tree);
    }
}
```

In 1 sentence explain why Nicks free function will cause problems on the following input. You may assume all calls to malloc succeed.

```
char *filename = malloc (sizeof (char) * (strlen("ex") + 1));
strcpy (filename, "ex");
simple_AST *tree = MakeAST (...., filename);
simple_AST *child = MakeAST (...., filename);
AppendAST (tree, child);
FreeAST (tree);
```

---

---

---

**Solution:** The code above will make two calls to free on the same pointer (filename). We can't free the same memory twice, this might crash our program!

helpful! | 0



**Jie Qiu** 4 months ago during the call FreeAST(tree), filename will first be freed by child (since FreeAST first frees all children), then it will be freed by tree again.

helpful! | 0



**Anonymous Poet** 4 months ago Is this because we recursively called free?

helpful! | 0



**Jie Qiu** 4 months ago yes. For each recursive call you free the filename that the current tree is pointing to.

helpful! | 0

Resolved  Unresolved

This was marked a duplicate to the question/note above by Stephan Kaminsky 4 months ago

""

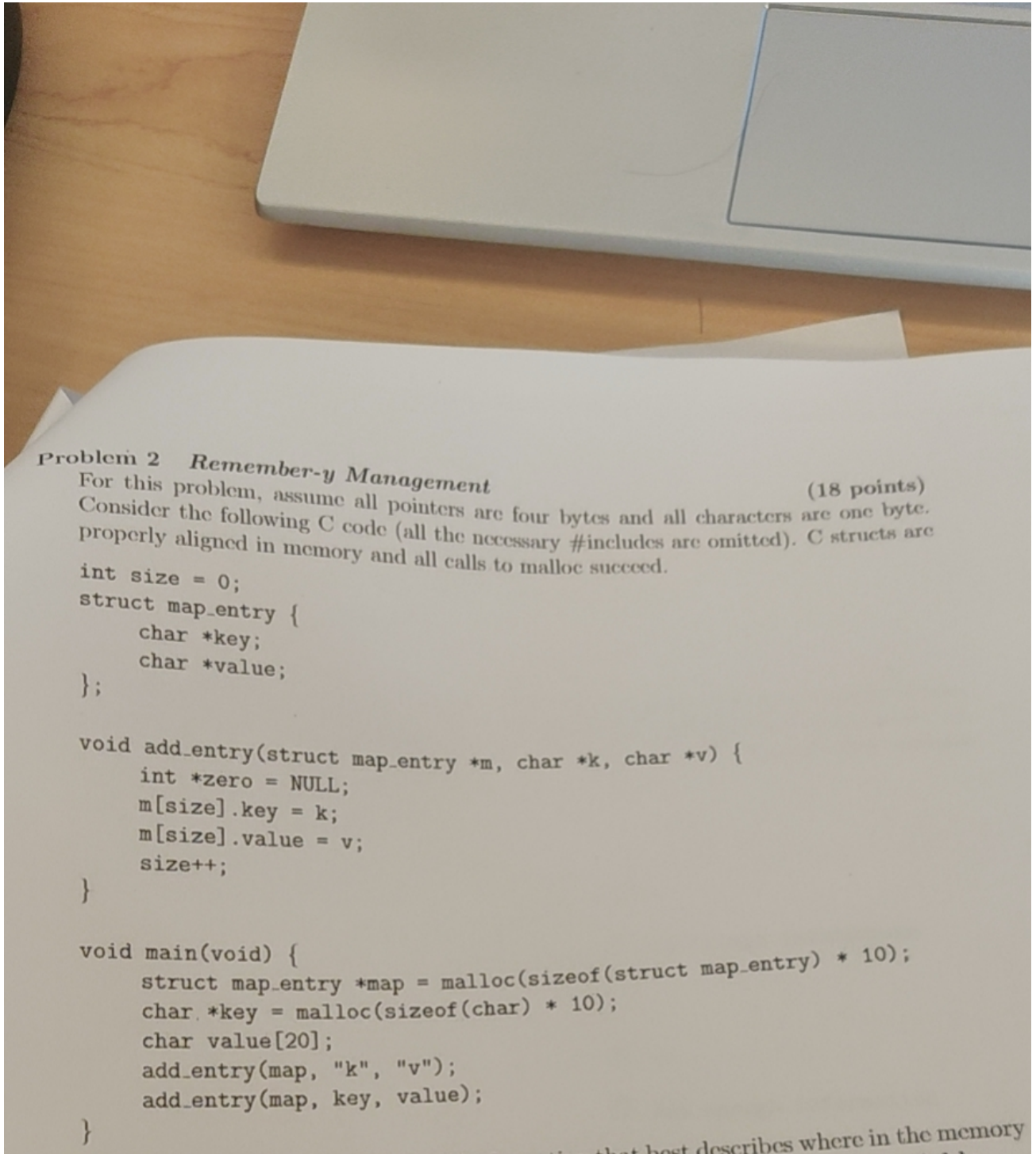


**Anonymous Calc** 4 months ago

**[SP-MT1]:Q2ac**

Heaps question

why is this in heap memory? I think heap refers to all memories that are explicitly malloced.



(a) For each of the following, bubble the option that best describes the memory layout each variable is stored in. You should select one answer per variable.

(a) zero

Stack

Static

Heap

Code

(b) `*map[0].key`

Stack

Static

Heap

Code

(c) `map[1].value`

Stack

Static

Heap

Code

MT 1

Page 7 of 22

CS61

CS61

helpful! | 0



**Jie Qiu** 4 months ago `map` is malloc'd, so all entries in `map` are on the heap. Thus, `map[1].value` is located in the heap but points to a stack address.

helpful! | 1



**Anonymous Comp** 4 months ago is it because it **WASN'T** dereference like the case with `*map[0].key`?

helpful! | 0



**Albert Magyar** 4 months ago Yes, exactly.

helpful! | 1



**[REDACTED]** 4 months ago Why is the dereferenced char array in static?



helpful! | 0



**Anonymous Gear 2** 4 months ago I think its because it's referring to the string literal, which is stored in static

helpful! | 0

Resolved  Unresolved



**Anonymous Atom** 4 months ago

[SP-MT1]:Q3

[SP-MT1]:Q5

[SP-MT1]:Q6

Is question 3 in scope?

I'm assuming problems 5 & 6 will not be in scope on this exam?

Thanks for the clarification (ps it would really help if a TA would be willing to go write the questions in scope in the past exam posts:)

helpful! | 0



**Stephan Kaminsky** 4 months ago Other than doing the prologue and epilogue from scratch with no guidance, 3 is in scope for MT1. 5 & 6 are not in scope for MT1.

helpful! | 0

Resolved  Unresolved



**Anonymous Poet** 4 months ago

[SP-MT1]:Q3

What does a0/s0 and s1/a1 mean?

```
# We can use a0/s0 and a1/s1 because  
# for now, they are the same values
```

helpful! | 0



**Anonymous Poet** 4 months ago Sorry this is [SP20]-{1}:Q{3}

helpful! | 0



**Jie Qiu** 4 months ago This means a0 and s0 hold the same value thus can be used interchangeably. So are a1 and s1

helpful! | 0



**Anonymous Poet** 4 months ago OH so we aren't dividing... just that either will work?

helpful! | 0



**Jie Qiu** 4 months ago Yes :)

helpful! | 0

Resolved  Unresolved



**Anonymous Poet** 4 months ago

[SP-MT1]:Q3

When we want to load something at the 4th bit, how do we stop it after loading bits 4, 5, 6, and 7? What is stopping it from loading the 8th bit and beyond as well?

# 0 = children, 4 = size, 8 = data  
# So load ast->data, and if it isn't

helpful! | 0



**Anonymous Poet** 4 months ago Same as above... [Sp20-1]:Q3

helpful! | 0



**Jie Qiu** 4 months ago I believe you are referring to **bytes** instead of bits. lw automatically loads in a word, which is 4 bytes, and nothing beyond will be loaded.

helpful! | 0

Resolved  Unresolved



**Anonymous Comp** 4 months ago  
[FA-MT]:Q1

I had a question regarding Fall 2019 Midterm for Question 1 part c (there was only a midterm and quest for that semester.

I understood how the answers for part a and b were determined but I am having a hard time following the explanation for part c.

I understand that for an exponent of  $2^7$  the step size is 1, but how is this true for numbers between 0 and 255? I thought the step size changes when the exponent changes?

### Part C

For an exponent of  $2^7$  with 7 mantissa bits you have a step size of  $2^{-7} * 2^7 = 2^0 = 1$ . This means every integer between 0 and 255 is representable.

helpful! | 0



**Jie Qiu** 4 months ago For numbers with an exponent less than  $2^7$ , the step size is going to be less than 1, thus every integer will be representable.

helpful! | 0



**Anonymous Comp** 4 months ago Thank you Jie!

helpful! | 0

Resolved  Unresolved



**Anonymous Helix** 4 months ago  
[SP-MT1]:Q2ed

(d) value `-- &zero`

>

==

<

Not enough information

Can somebody explain why its ">"?

helpful! | 0



**Anonymous Comp** 4 months ago Value is on the STACK whereas &Zero is on the stack but it was called twice AFTER Value was initialized.

helpful! | 0



**Evan Sum** 4 months ago Something to note is that the stack grows downward. This means frames and variables initialized later are lower in memory. So, as the Anon said above, zero is declared after value was declared and value points to somewhere on the stack, giving value a higher value than &zero

helpful! | 1

Resolved  Unresolved



**[REDACTED]** 4 months ago

**[SU-MT1]:Q1.3**

It said that the items in a product group is not in a same size that it may be between 1 and 15. So, I think the answer should depend on the details of how many items each group actually has because we can use flexible prefix to encode it in some situation. For example, there is only one group has 15 item and other groups have only one item. In this case, I can assign the group number of the 15-items group as 11, and 00000, 00001, 00010, ..., 01010, 01011 for the rest eleven groups which only have only one item in it.

3. We expand to have 12 product groups. The largest has 15 items in it while the smallest has one item. Nick argues the entire barcode can now be condensed to only 6 bits without losing product grouping or unique identifiers. Is he correct? If yes, explain why, if no, what is the actual minimum size?

Yes, he is correct

No, he is incorrect

At least 4 bits for items ceiling(log2(15))

At least 4 bits for groups ceiling(log2(12))

So it can't be less than 8 bits

helpful! | 0



**Jie Qiu** 4 months ago That might be a viable solution, but I think this question assumes the bit breakdown is fixed. Also, it can be difficult to generalize the flexible prefixing when your numbers of product groups and products can change.

helpful! | 0

Resolved  Unresolved



**Anonymous Comp** 4 months ago

**[SP-MT1]:Q2**

I had a question regarding Question 2 from Spring 2019 Midterm 1 Exam. Based on the description, it says "assume malloced memory grows upward." However, I am not sure what to say in the case if the questions asked about **\*map** and **\*key**? These would be on the heap, but if malloced memory grows upward on the stack, does this apply to whats on the heap?



(c) `map == key`

`>`

`==`

`<`

Not enough information

helpful! | 0



**Jie Qiu** 4 months ago The question statement is correct. You are comparing the values of `map` and `key` which are just memory addresses.

helpful! | 0



**Anonymous Comp** 4 months ago Sorry Jie, I did not explain my question clearly. I understand the question and answer.

I am asking what if the question had asked us regarding `*map` and `*key` instead. What would the answer have been in this case?

helpful! | 0



**Jie Qiu** 4 months ago In that case, `*map` is a struct while `*key` is a char, so I don't think it is valid to compare them.

helpful! | 1



**Anonymous Poet 2** 4 months ago when comparing between memories. Are the bottom locations(static and code) greater than heap and stack?

helpful! | 0



**Anonymous Poet 2** 4 months ago like why on key vs `&size` is key greater. If key is in the heap and `&size` located in static?

helpful! | 0



**Jie Qiu** 4 months ago Bottom locations have lower addresses. So `stack > heap > static > code`

helpful! | 0

Resolved  Unresolved



**Anonymous Comp** 4 months ago  
[SP-MT1]:Q2

I had a minor question regarding Question 4 from Spring 2019 Midterm 1 Exam. This is for parts a-d).

When the question ask to NOT simplify, the answers, did it mean not to put in in scientific notation? On the exam, if a question was asked like this, should we assume it meant to have a decimal answer?

helpful! | 0



**Zac Patel** 4 months ago Not to simplify basically means don't spend the time to multiply it out. For example, on 1a) leaving it in  $3200 \times 10^6$  is fine since that is - we would have given full credit to answers like  $3.2 \times 10^9$  for example but it was not required.

For most questions you should expect to see on tests, leaving it as an unreduced fraction or in scientific notation is fine unless otherwise stated.

helpful! | 0

Resolved  Unresolved



**Anonymous Comp** 4 months ago

**[SU-MT1]:Q4**

I had a question regarding Question 4 Part 2 from Summer 2019 Midterm 1 Exam.

In the solution, it specifies to dereference the char\* text for a caption\_t using "." and not "->". Why is this the case? Isn't caption\_t\* array pointing to the caption\_t struct?

```
int longest_caption(video_caption_t* video) {
    int max_len = 0;

    for (int i=0; _____ i<video->length_____ ; i++) {

        _____int length = strlen(video->caption_array[i].text);_ ; // be careful of -> and . here
    }
}
```

helpful! | 0



**[REDACTED]** 4 months ago If look at the code right above, where the video\_caption\_t struct is defined it says "caption\_t\* array; //pointer to "length" consecutive caption\_t struct". I think this implies that caption\_t\* array is pointing to a block of memory where consecutive caption\_t's are stored as opposed to pointing to other pointers. Otherwise it's type would be "caption\_t\*\* array". When you move the array pointer along, you are just moving which caption\_t it is pointing at.

helpful! | 0

Resolved  Unresolved



**Anonymous Comp** 4 months ago

**[SU-F]:Q1**

I had a question regarding Final Exam Summer 2019 for Part 8. I think there is a mistake in the answer key here because there should only 16 numbers not 16368. Every power of 2 in the exponent means 16 numbers can be represented with the mantissa.

Am I correct here?

8. How many numbers does our floating point scheme represent in the range  $[0, 1)$  (the range 0 to 1, where 0 is included and 1 is not)? For this question assume  $-0$  is not in this interval. You may leave your answer unsimplified.

1 has an exponent value of 1023 because  $2^0 = 1$ . This means all the values with a positive sign and exponent less than 1023 are in this range. Because there are 4 significant bits there are 16 numbers per exponent.

$$= 1023 * 16 = 16368$$

helpful! | 0



**Jie Qiu** 4 months ago I don't quite understand what you are asking here, but the idea is that 1 has an exponent of 1023. Thus, all numbers with exponent less than 1023 are in range  $[0, 1)$ . For every exponent in range  $[0, 1022]$ , the mantissa bits can have 4 representations. The answer is therefore  $1023 * 16$

helpful! | 0



**Anonymous Comp** 4 months ago I understand better now!

helpful! | 0

Resolved  Unresolved



**Anonymous Comp** 4 months ago  
[SP-F]:Q4

I had a question regarding the Final exam during Spring 2019 for Question 4. In regards to Floating Point scheme, I couldn't find anywhere in the question where they specified how the Exponent value is determined? How is the value of the Exponent incorporated in this scheme? Or were they trying to say the Exponent is 127 as a constant?

#### Question 4: **B**loating Point (9 pts)

Being the feisty floating point fanatic you are, you devise a new scheme to interpret 32-bit floats. You keep the breakdown of the Sign/Exponent/Significant fields the same. However, there is no implicit 1. or 0. before the significant bits; also, you evaluate the 23 significant bits as an **unsigned number**, which you then multiply with  $2^{Exp - Bias}$  and  $(-1)^{Sign}$  as you normally would.

$$(-1)^{Sign} \times \text{Significant}_{\text{unsigned}} \times 2^{Exp - Bias}$$

where  $Bias = 127$ .

There are no denormalized numbers.

Exponent	Significant	Value
Largest	Zero	$\pm$ Infinity
Largest	Nonzero	NaN

We walk through one way of representing 3 in this scheme, using a Significant of  $0b00\dots011 = 3$   
 $(-1)^0 \times 3 \times 2^{127 - Bias} = 1 \times 3 \times 2^0 = 3$  **Sign:** 0, **Exponent:** 127 = 0x7F, **Significant:** 3 = 0x000003

helpful! | 0



**Jie Qiu** 4 months ago It's the same as IEEE 754 standard.

helpful! | 0



**Anonymous Comp** 4 months ago but if it the same as IEEE, doesn't that make the exp component irrelevant?

helpful! | 0



**Jie Qiu** 4 months ago Irrelevant to what?

helpful! | 0



**Albert Magyar** 4 months ago I edited your tag to properly represent that this is a final exam question.

helpful! | 0



**Albert Magyar** 4 months ago You take the value of the exponent field in the biased system (which means subtracting the bias), and raise two to that power. This term is multiplied by the mantissa value, as spelled out explicitly in the question.

helpful! | 0

Resolved  Unresolved



**Anonymous Mouse** 4 months ago

**[FA-Q]:Q3**

This is probably dumb, but why is `int32_t *3` only allocating 12 bits? Does `int32` mean we are on a 32 bit system so each `int` is 4 bits, or does it mean that the `int` is 32 bits? Or have I just been thinking about it all wrong--does a 32 bit system have 4 bit pointers, and not 4 bits for ints? Can someone please clarify

helpful! | 0



**Jie Qiu** 4 months ago it allocates 12 bytes because each `int32_t` is 32 bits (4 bytes). On a 32 bit system, pointers are 32 bits (4 bytes). 1 byte = 8 bits.

helpful! | 1

Resolved  Unresolved



**Anonymous Scale** 4 months ago

Um, I tried to make a video for Spring 2019 Midterm 1 to see if this helps.

[https://www.youtube.com/watch?v=ZxMfbSXJS0U&list=PLXdl-bKzbrT5qIMUdt3\\_M8FA764eIIWdG](https://www.youtube.com/watch?v=ZxMfbSXJS0U&list=PLXdl-bKzbrT5qIMUdt3_M8FA764eIIWdG)

~ An instructor (Jie Qiu) thinks this is a good comment ~

helpful! | 7



**Anonymous Comp** 4 months ago thank you Evan! :D

helpful! | 0

Resolved  Unresolved



**[REDACTED]** 4 months ago

**[SP-MT1]:Q3**

In `lftwo`, instead of "`jalr ra a2/s2 0`", can we use "`jal s2`" to call the function stored in `s2`?

helpful! | 0



**Evan Sum** 4 months ago So, we can't because we actually need to store the location that we are currently at before jumping to the function call. Let's say, for example that you were to use `jal s2`. Then, after calling the function `f` onto `data`, we would return or jump back using `ra` right? But if we

didn't save the current ra that you're talking about, we'd actually jump back to the function that called searchAST.

original ra = return address that searchAST should be.

`jalr ra a2/s2 0`

does whatever s2 is (which is the function call f)

uses ra to jump back and continue to the for loop.

-----

if we do `jal`

original ra = return address that searchAST should be.

~~jal s2~~ my bad! I forgot that `jal` must be `label`. Anonymous Comp is correct. You must do either `jalr` or `jr` since you only have access to the function through a register

~~does whatever s2 is (which is the function call f)~~

~~use ra to jump back and continue to the for loop, and leave searchAST on accident.~~

helpful! | 0



[REDACTED] 4 months ago That's very clear. Thanks!

helpful! | 0



[REDACTED] 4 months ago Could you do `jal ra a2` instead of `jalr ra a2 0`?

helpful! | 0



Anonymous Comp 4 months ago I think there is a misunderstanding here.

We have to use **jalr** because the address is in a register. So we can't use **jal**

Also **Jal <label>** is the exact same as **Jal ra <Label>**

helpful! | 1



Anonymous Comp 4 months ago

$$j < label > \equiv jal\ x0, < label >$$
$$jal < label > \equiv jal\ ra, < label >$$

helpful! | 0



[REDACTED] 4 months ago That clears things up thanks! Also does the 0 act as the offset? So since we just want the function within the register we don't want any offset?

helpful! | 0



Evan Sum 4 months ago Sorry about the confusion, I added a comment up top.

To answer your question, yes. The function is stored in s2, and exactly there. So, we want to jump exactly there with no offset.

helpful! | 1



[REDACTED] 4 months ago Very helpful, thanks!

helpful! | 0

Resolved  Unresolved



**Anonymous Beaker** 4 months ago  
[SP-MT1]:Q2

I am a little confused about question 2e.

When would the answer be ==? Is it only when we are comparing two identical variables or different variable pointers pointing to the same thing?

Similarly, I am a little confused as to why the key is on the heap and not on the stack. Is it because the key is on the heap, but the thing it points to is on the stack? By this logic, would &key be on the stack while key would be on the heap?

helpful! | 0



**Jie Qiu** 4 months ago The answer would be == if the two pointers we are comparing are pointing to the same thing. key's value is a heap address, but key itself is on stack. So yes, key is heap and &key is on stack

helpful! | 1

Resolved  Unresolved



**Anonymous Calc 2** 4 months ago  
[SP-MT1]:Q2

(a) map -- zero

>

==

<

Not enough information

I'm confused on this question. I thought zero is on the stack while map is on the heap so shouldn't the map be < than zero?

helpful! | 0



**Evan Sum** 4 months ago Hello! you are correct in which map points to the heap, such as the map contains the value of something in the heap. However, zero is on the stack, in which the ADDRESS of zero is on the stack. However, zero itself has the VALUE 0. While part 1 of this question was asking about addresses, part 2 of this question was asking about values.

helpful! | 1



**Anonymous Calc 2** 4 months ago I see! Thank you for clearing this up!

helpful! | 0

Resolved  Unresolved



**Anonymous Atom 2** 4 months ago  
[SP-MT1]:Q4

Is Question 4 on Spring 2019 Midterm 1 in scope? I feel like we did not cover Kibibytes or IEC prefixes. Am I missing something perhaps?

helpful! | 0





**Jie Qiu** 4 months ago It's covered in discussion 1! Also you can find all the information you need about IEC prefixes on the green card.

helpful! | 0

Resolved  Unresolved



**Anonymous Comp 2** 4 months ago

[SP-MT1]:Q2e

Why is the number of bytes leaked 90? I thought the struct is 4 bytes (2 bytes of padding) \* 10 for the map malloc and 1 byte \* 10 for the key malloc so together thats 50 bytes?

helpful! | 0



**Evan Sum** 4 months ago Hello! So this is one of the ways they tried to trick you. Keep in mind that char \* is a pointer, which is 4 bytes each. That means the struct is actually 8 bytes.

helpful! | 2

Resolved  Unresolved



**Anonymous Helix 2** 4 months ago

[SP-MT1]:Q4

On an exam, if we wrote 210 for 2a instead of 0dTCA, would we still receive full credit?

For problems like this should we assume to write the letters?

helpful! | 0



**Evan Sum** 4 months ago Hello! So I took this exam and I can confirm that you could write either. They were a bit unclear in the instructions. I think you should ask for a clarification.

### QUESTION 22

4.2.2 (b)

2 / 2 pts

✓ + 2 pts Correct: 1330 or CGGA

+ 0 pts Incorrect

helpful! | 0

Resolved  Unresolved



**Anonymous Comp** 4 months ago

[SP-MT1]:Q3

I had a question regarding Question 3 from Spring 2019 Midterm 1. I was a bit confused how to handle the pointer addition to get `lst->children[i]`.

I understood the solution up until the command line that did the pointer addition

```
add t2 t1 t0
```

So we obtained the address stored in the double pointer (children); this corresponds to the code that says `lw t0 0(s0)`. This address that points to another pointer (contains an address) which points to another struct.

I understand the `i` needs to be incremented to get to the specific index in children but I am having a hard time making sense of adding the address stored in children and adding it after the incrementation of `i` gets us `ast->children[i]`.

How does the pointer addition make sense ?

```
Loop:      # Check our children
           lw t0 4(s0) # Load ast->size
           bge s3 t0 Done # if i >= size we return
           lw t0 0(s0) # load ast->children
           slli t1 s3 2 # t1 == i << 2 to create byte offset to i'th child
           add t2 t1 t0 # Pointer addition to get &(ast->children[i])
```

helpful! | 0



**Evan Sum** 4 months ago Hello! So hopefully this example can clarify things.

Let's say `ast` is stored at location `0x50000000` with a value (which is a pointer) of `0x60000000`, so `s0=0x50000000`. This also means that the first child, `ast->children[0]`, would be found from `0x60000000 - 0x60000003`. The second child, `ast->children[1]`, would be found from `0x60000004 - 0x60000007` and so on.

That means that the current `0x50000000 - 0x50000003` contains the 4 byte pointer (`0x60000000`) to `ast->children`.

Now, what might help here is that when we do a `lw`, we are actually getting the VALUE at the given pointer, similar to dereferencing.

So after `lw t0 0(s0)`, `t0` now contains the VALUE of whatever was stored in `0x50000000 - 0x50000003`, which is `0x60000000`. `t0 = 0x60000000`

Lastly, we do the bit shifting to get the index just like you said, and we add that to `t0`, giving `t0` the correct location for `ast->children[i]`.

LMK if you have any questions.

helpful! | 0

Resolved  Unresolved



**[REDACTED]** 4 months ago

**[FA-MT1]:Q3**

is 1s compliment in-scope?

helpful! | 0



**Anonymous Comp** 4 months ago Yes

helpful! | 0



**[REDACTED]** 4 months ago do you know which lecture did we talk about sign/magnitude?

helpful! | 0



**Daniel Li** 4 months ago Sign and magnitude is not covered this semester but sign and magnitude just takes the first bit as the sign (1 if negative, 0 if positive) and the rest of the bits as the magnitude.

helpful! | 0

Resolved  Unresolved



**Anonymous Mouse 2** 1 month ago

could we possibly get grade distributions/statistics for last spring's final?

helpful! | 0



**Stephan Kaminsky** 1 month ago We are sorry but we will not disclose that.

helpful! | 0

Resolved  Unresolved



**Anonymous Scale 2** 1 month ago [SP-F]:Q9

Can someone explain the solution. I don't understand how it was derived at all.

- a) I picked XOR->XOR->OR as the critical path, why include the and gate?
- d) and the rest of the circuits, how would you use boolean algebra to solve this problem?
- g) Why B? Why do any of the signals have undefined behavior?

helpful! | 0



**Daniel Zhang** 1 month ago a) You include the AND gate because there also is a path from the first XOR to the second XOR that goes through AND, and we want to find the longest possible path in the circuit. Using 170 terminology– the critical path is the longest path in the directed acyclic graph from the register at clock cycle 1 back to the same register at clock cycle 2. If there's a way to increase the time taken in a path, then that path is not a critical path.

d) What I typically do is draw out truth tables. I do believe, though, that there was a discussion problem on a previous discussion that covered really briefly creating other logical circuits out of NAND circuits– it's a good thing to have on your cheat sheet so that you don't have to waste precious minutes trying to rederive during the exam.

g) The signals have undefined behavior because they're reliant on input data from we do not have access to (e.g., before time 0). B matches the circuit as seen right after the third sub-part– if you draw out the timing diagram, you'll see that A, C, D all have multiple parts that don't match the correct output.

helpful! | 0

Resolved  Unresolved



**Anonymous Beaker 2** 1 month ago

[SU-MT2]:Q ???

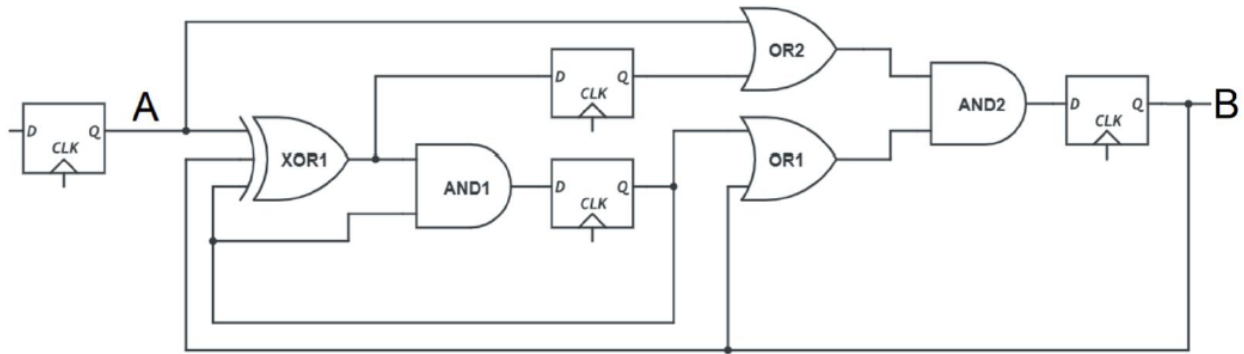
I had a question regarding a problem from Summer 2019 Midterm 2. I had a hard time understanding the explanation the answer wrote. I understood the derivation for the critical path.

Why does it make sense, at least how the answer was intended to be, for the minimum latency to be 2x the critical path and not just the longest pathy (longest time) it takes to get to go from A to B? In other words, it would be the critical path plus the gates/register right before it as well.

### SOLUTIONS Summer 2019 Midterm 2 (cont.)

For the next problems, consider the following pipelined circuit. Assume all registers have their clock inputs correctly connected to a global clock signal and that logic gates have the following parameters:

- XOR gate delay: 80 ps
- AND gate delay: 60 ps
- OR gate delay: 40 ps



When shopping for registers, we find two different models and want to determine which would be best for our circuit.

#### Register Type $\lambda$

- Setup Time: 40 ps
- Hold Time: 20 ps
- Clock-to-Q Delay: 30 ps

#### Register Type $\tau$

- Setup Time: 10 ps
- Hold Time: 10 ps
- Clock-to-Q Delay: 80 ps

helpful! | 0



**Charles Hong** 1 month ago The propagation delay from A to B can't just be the sum of the critical paths, because all the registers share the same clock. Registers don't propagate their input until the rising edge of the clock, so even if it gets its input faster than the clock period, it can't propagate that input to its output until the next rising edge of the clock.

So, we must first determine the critical path of the circuit (to make sure there are no setup time violations), then determine the clock period, then see that in order to fully propagate A to B, it will take 2 clock cycles, since it will take a full cycle for the output of the middle registers to change, then another full cycle for the register on the right to update based on the middle registers..

helpful! | 0

Resolved  Unresolved



**Anonymous Poet 3** 1 month ago

[SP-F]:Q6

The answer is Line 6/7/11/13.

Can someone help me explain why there are errors and how to correct it?

Thanks!!

### Problem 6 C Reading

(16 points)

The function `parse_message` takes two inputs: an array of strings, and the length of the array. It copies the strings from the input array into a new buffer, ending the buffer with a NULL ptr rather than specifying a size. However if any of the strings are the string "STOP", then it terminates early and returns only strings before the stop message, again ending with a NULL terminator.

- (a) The function below contains *at most 5 bugs* which cause the function to non-deterministically exhibit incorrect behavior. Bubble in the lines of code that may produce errors. You may select more than one line.

You may assume all calls to `malloc` succeed, `arr` and its contents are never NULL, `arr` always has at least `size` allocated, and we are using C99.

```
 1. char** parse_message (char** arr, size_t size) {
 2.     int init_size = 8;
 3.     char **output = malloc (sizeof (char *) * init_size);
 4.     int i;
 5.     for (i = 0; i < size; i++) {
 6.         char *pointer = * arr + i;
 7.         if (pointer == "STOP") {
 8.             break;
 9.         } else if (init_size == i - 1) {
 10.             init_size *= 2;
 11.             realloc (output, sizeof (char *) * init_size);
 12.         }
 13.         output[i] = malloc (sizeof (char) * strlen (pointer));
 14.         strcpy (output[i], pointer);
 15.     }
 16.     output[i] = NULL;
 17.     return output;
 18. }
```

helpful! | 0



**Jie Qiu** 1 month ago Line 6 should be `char *pointer = arr + i` since pointer is a `char *`, we do not want to dereference it.

Line 7 you should use `strcmp` as `==` compares the addresses (since strings are just char arrays)

Line 11 realloc might fail

Line 13 you need to allocate one more byte for the null terminator.

helpful! | 1



**Daniel Fan** 1 month ago To add onto Jie's response,

`realloc` returns a `void *` so you actually need to do `output = realloc` as `realloc` may have moved the buffer in order to increase its size.

it's left out on the sols for some reason, but line 9 is also incorrect—it should be `init_size - 1 == i`

helpful! | 0



**[REDACTED]** 1 month ago For "Line 6 should be `char *pointer = arr + i` since pointer is a `char *`, we do not want to dereference it."

arr is a pointer to char\* and pointer is a char \*, so I think we should use pointer=\*(arr+i) or arr[i]. Why we should use char \*pointer = arr + i, as the left side is a char\* and right side is a char\*\*?

[helpful!](#) | 0



**Daniel Fan** 1 month ago You're correct. It should be \*(arr + i) or arr[i]. Good catch.

[helpful!](#) | 0

Resolved  Unresolved



**Anonymous Calc 3** 1 month ago

[SP-F]:Q2a

Why isn't C a valid answer as well?

[helpful!](#) | 0



**Anonymous Calc 3** 1 month ago Nevermind, I believe it's because they specified least amount of hardware.

[helpful!](#) | 0

Resolved  Unresolved



**Anonymous Calc 3** 1 month ago

[SP-F]:Q3

How did they get  $2^{10}$  pages?

[helpful!](#) | 0



**Anonymous Calc 3** 1 month ago [SP-F]:Q3e

[helpful!](#) | 0



**Jie Qiu** 1 month ago We have  $2^7$  bytes/page,  $2^{22}$  bytes of virtual memory, thus  $2^{15}$  pages. Each page takes up 4B in the page table. So the page table is  $2^{17}$  bytes,  $2^{10}$  pages

[helpful!](#) | 1

Resolved  Unresolved



**Anonymous Calc 3** 1 month ago

[SP-F]:Q4c

Is the new code slower due to less parallelism?

[helpful!](#) | 0



**Daniel Zhang** 1 month ago Yup! That's correct. It's because in the reduction code we don't constantly access s1 from multiple threads like we do in part (c).

[helpful!](#) | 0

Resolved  Unresolved



**Anonymous Atom 3** 1 month ago

[SP-MT2]:Q6b



How did they arrive at 627/800? It seems a bit too high, as I thought a direct-mapped cache will force the cache to bounce between storing either products or reviews items in a block as they continuously overwrite each other, since each loop makes you first grab an item from products, then grab an item from reviews.

helpful! | 1



**Anonymous Comp 3** 1 month ago +1 I had the same logic since in the index is the same. I thought you would start getting a hit when  $i = 0$ ,  $j = 16$  since both products and reviews will pingpong on index 0 back and forth until the index of reviews would be  $0x04000010$ ? I'm not too sure though. Can an instructor elaborate?

helpful! | 0



**Daniel Fan** 1 month ago This was covered in one of the dead week review sessions :D Here's a link: <https://www.youtube.com/watch?v=8AL98DeRoG8&list=PLDoI-XvXO0apoG8B70j3H6gpz8aDDaaNR&index=16&t=0s> I believe the question is the last one in the video.

helpful! | 1



**Anonymous Comp 3** 1 month ago I still dont get it why `review[4]` maps to a new block. If we follow the memory address for review, `R[4]` lives in  $0x04000004$ , which still has the same Index as product? Also since its a char array, everytime we miss dont we store up to 16 characters? How can `R[8]`, `R[12]` be a miss?

helpful! | 0



**Anonymous Atom 3** 1 month ago Why is it 4 items per block? As each character is 1B, and the block size is 16B, how does it miss at every 4? How would we have known that each character takes up 4B? I think this is what confused me originally.

helpful! | 0



**Daniel Fan** 1 month ago Array of char pointers, each pointer is 4 bytes.

helpful! | 2



**Anonymous Atom 3** 1 month ago Thank you!

helpful! | 0



**Anonymous Comp 3** 1 month ago Thank you for saving me 2 hrs of my life when i had the correct logic but thought it was char LOL

helpful! | 0



**Daniel Fan** 1 month ago It happens to everyone haha

helpful! | 0



**Anonymous Comp 3** 1 month ago Real talk tho, did anyone have that much time to do this question at all? Just wondering since the difficulty of this question is through the roof.

helpful! | 0



**Daniel Fan** 1 month ago The general consensus on the resulting piazza thread seemed to indicate that most (if not all) people did not have time to get the answer for that question. Anecdotally, I gave up and guessed on the question and still did pretty well relative to the curve, so make of that what you will.

helpful! | 2



**Anonymous Comp 3** 1 month ago

**[SP-MT2]:Q6b**

How did they get 19/20? So my logic is:

p[0] miss r[0] miss

p[0] hit r[1] miss (and overrides r[0] since its a 4B block)

p[0] hit r[2] miss and etc.

So the first iteration has 21 misses? And the cache should have p[0] at index 0 with r[3] and r[7], r[11], r[15], r[19] at index 1 -4 respectively

Product on the outer loop has to miss another 19 times so thats already 40 without even counting the inner loops of each of the outer loop which should be higher?

helpful! | 0



**Anonymous Comp 3** 1 month ago 6c\*

helpful! | 0



**Daniel Fan** 1 month ago The cache is big enough to hold everything, so the only misses are the compulsory ones in bringing the blocks in for the first time.

helpful! | 1

Resolved  Unresolved



**Anonymous Helix 3** 1 month ago

**[SU-F]:Q4**

"Translate the body of mystery from the handout from C to RISC-V. "

Where's this mystery code, what handout? I cant seem to find it on either the blank exam or solutions :(

helpful! | 0



**Sunay Poole** 1 month ago

[https://drive.google.com/file/d/1RLGSPKcFzUVFOKZrBm\\_B7ay0xC1d1I6W/view](https://drive.google.com/file/d/1RLGSPKcFzUVFOKZrBm_B7ay0xC1d1I6W/view)

helpful! | 1

Resolved  Unresolved



**Anonymous Gear 3** 1 month ago

**[FA-F]:Q3**

I didn't realize that if you malloc a pointer for an array of a particular objects, it automatically initializes each of the objects.

Like in the for loop, we could just assume that the L->edges[L->len] had an A and B attribute.

Is this a reasonable interpretation of what is happening here? Just kinda confused.

helpful! | 0

Resolved  Unresolved



**Anonymous Atom 2** 1 month ago

**[SP-F]:Q3b**

For question 3b, how did they find the immediate that represents "Cont"?

helpful! | 0



**Anonymous Atom 2** 1 month ago Never mind, figured it out. They added 8 to the current PC counter.

helpful! | 0



**Anonymous Atom 2** 1 month ago This is for Spring 2018 by the way

helpful! | 0

Resolved  Unresolved



**Anonymous Mouse 3** 1 month ago

**[SU-F]:Q9.2**

I understand why A is correct but I'm not sure why B is an incorrect answer. From what I can see, it does the correct action of choosing PrevALUOut when EXEXFwd=1 and does the normal ASel process otherwise. Therefore, it seems like their outputs would be the same, so why is B wrong?

helpful! | 0



**Sunay Poole** 1 month ago Give me til tomorrow, I'm making a walkthrough video for this exam right now lol. But yeah for that question, B was also accepted as a correct answer, the solutions just weren't updated.

helpful! | 0



**Anonymous Mouse 3** 1 month ago Oh okay, cool thanks!

helpful! | 0

Resolved  Unresolved



**Anonymous Gear 3** 1 month ago

**[SP-F]:Q3b**

So I am confused by the value that they gave for the size of a Page Table Entry: 4B.

Doesn't the page table map VPNs to PPNs which are 15 and 27 bits respectively. How then could 15 + 27 bits fit into a 32 bit entry?

I'm clearly missing something. Does the page table not store the entire VPN or PPN?

helpful! | 0



**Daniel Fan** 1 month ago The VPN is implicitly "stored" because it is used as the index into the page table. If your VPN is say 13, you would access the 13th PTE entry. 32 bits is enough to hold 27 bits + some status bits so 4 bytes is what you'll typically see in practice.

helpful! | 1



**Anonymous Gear 3** 1 month ago Ah that makes perfect sense. Thanks.

helpful! | 0

Resolved  Unresolved



**Anonymous Gear 3** 1 month ago

**[SP-F]:Q3d**

I didn't understand why the access pattern needed to traverse any more than 1 page, since 64B for the page table would fit into one page.

Is it because we need to access the page table base register to get the address of the page table?

And so is page table base register not a register at all but just also in DRAM?

helpful! | 0



**Daniel Fan** 1 month ago PTBR is a register; it is not stored in DRAM. I went over my reasoning for why it would need to traverse at least 2 pages in the video walkthrough. Which part of it are you confused about in particular?

helpful! | 0



**Anonymous Atom 3** 1 month ago On a separate note, could I get a link to the video walkthrough? Thanks!

helpful! | 0



**Daniel Fan** 1 month ago It's already listed above as an update to this post.

helpful! | 1



**Anonymous Atom 3** 1 month ago oh shit, did not see that, my brain is little bit fried rn. Thank you!

helpful! | 0

Resolved  Unresolved



**Anonymous Comp 3** 1 month ago

[SP-F]:Q5

Will we be given the cheat sheet? I cannot see how we could do the question given how big intel's SIMD library is (if we were to choose since its open book)

helpful! | 0



**Daniel Fan** 1 month ago For those kind of questions, you'll always be provided with a list of useful SIMD functions.

helpful! | 0

Resolved  Unresolved



**Anonymous Helix 3** 1 month ago

[SP-MT2]:Q6a

What is the logic behind the answer to this problem? The associativity is increased so it makes sense conflict should decrease, and block sizes smaller so compulsory should increase, but how would we distinguish between top left & bottom right answers?

helpful! | 0



**Daniel Fan** 1 month ago The official reasoning is that one of them adds up to 100% while the other doesn't >.<

helpful! | 0



**Anonymous Helix 3** 1 month ago haha nice thanks!

helpful! | 0

Resolved  Unresolved



**Anonymous Scale 3** 1 month ago

[SU-F]:Q6

5. Provides a performance improvement (i.e. faster read and write operations)

A Striping

B Mirroring

C Both

D Neither

Striping makes read and write operations better. Mirroring also improves reads without harming writes.

Why not striping faster than Mirroring? Since Mirroring is more reliable, it should be slower.

helpful! | 0



**Daniel Fan** 1 month ago You're right, striping does provide more write bandwidth than mirroring. However, the question only asks which options provide an improvement—not, which option provides the greatest improvement. And mirroring does increase your read bandwidth as you can issue reads simultaneously to a disk and its mirror.

helpful! | 1

Resolved  Unresolved



**Anonymous Poet** 1 month ago

[FA-F]:Q7cd

```
addi s0 x0 4
slli t1 s0 2
bge s0 x0 greater
xori t1 t1 -1
addi t1 t1 1
```

greater:

```
mul t0 t1 s0
```

In this case, why isn't there a data hazard? We are changing the value of s0 in the first line right? And then we are using that new value of s0 to slli the t1, but wouldn't there be a data hazard in that the slli will look at the old value of s0 instead of the new?

helpful! | 0



**Sunay Poole** 1 month ago The question says "after fixing that hazard, the following case fails", implying that we have already dealt with the data hazards from parts a and b and don't need to worry about hazards of this form anymore.

helpful! | 0

Resolved  Unresolved



**Anonymous Comp 3** 1 month ago

[SU-F]:Q6

Is there an error in the solutions? How can be parity bit P4 be equal to 0 when the fourth bit position of the data string is clearly 1?

helpful! | 0



**Anonymous Comp 3** 1 month ago Whoops I didnt read that it was Odd Parity instead of even.

helpful! | 0



**Sunay Poole** 1 month ago Yup, reminder that there's a video walkthrough linked in the post if it helps!

Resolved  Unresolved

**Anonymous Beaker 3** 1 month ago  
[SP-F]: Q2e

Under my calculation I used RF read to get  $30+150+160+75 = 415$  - it seems like in the answer they don't include a RF read in this stage -- why is that??

helpful! | 0



**Daniel Fan** 1 month ago Hi,

I addressed this in the video walkthrough above. Lmk if my thought process wasn't clear enough.

helpful! | 0



**Anonymous Beaker 3** 1 month ago Hi,

It made sense for the most part, but you mentioned that the RF would have its own RF read, and part 1 was the regfile, so shouldn't there be an added RF reader in the second component of the pipeline, resulting in  $30+160+150+75$ ?

helpful! | 0



**Daniel Fan** 1 month ago If you look at Part A, you'll notice that Block 1 is the modified register file. It would not make sense to include the time the original register file would have taken when considering the critical path of the new modified pipeline.

helpful! | 2

 Resolved  Unresolved

**[REDACTED]** 1 month ago



[SP-F]:Q3

(d) How many unique NON-DATA, NON-CODE pages does this access pattern traverse? (ie. page table pages)

- 0 pages
- 2 pages
- 3 pages
- 8 pages
- 12 pages
- 13 pages
- 16 pages

(e) Suppose this process has a single-level page table that starts out empty and we run through this access pattern. How much space would the page table take in memory? Give your answer in units of PAGES (*NOT BYTES*).

Size = \_\_\_\_\_ pages.

**Solution:**  $2^{10}$  pages

For (d), I know that one of the 2 pages is the page table page, and since one page contains 32 PTE, and the data pages are only 16 pages in total, the other page in this problem is not page table page. What is it?

For (e), why there are  $2^{10}$  pages in memory. I think that the virtual memory has 22 bit, the offset inside the page is 7 bits, so the number of virtual page is  $2^{(22-7)}=2^{15}$  pages. Where am I get wrong?

Thanks.

helpful! | 0



**Daniel Fan** 1 month ago Hi,

I addressed both of these in the video walkthrough above. Lmk if my thought process wasn't clear enough.

helpful! | 0

Resolved  Unresolved



**Anonymous Poet 4** 1 month ago

[SP19-F] 7.f.2

## 2. The imm in beq x0 x1 LABEL gets replaced with its final value

Compiler

Linker

Assembler

Loader

Why can we guarantee that LABEL is not an external label? Why is it for jal we recognize the possibility of an external jump, but with branches we assert that they must always be within the same file? For this reason shouldn't linker be a possibility too?

helpful! | 0



**Charles Hong** 1 month ago

Yes, we assume that `beq` doesn't jump to external files. I'm not sure if this was covered in class in depth at all, but I assume that the reason for this is that compilers usually only output `beq` for things that don't make references to external files, like `if` statements and loops. On the other hand, compilers use `jal` and `jalr` instructions for function calls, which can make reference to external files.

helpful! | 1

Resolved  Unresolved



**Anonymous Calc 4** 1 month ago

**[SP19-F] Q3**

Could someone please explain how they got these numbers?

### Problem 3 *Virtual Memory*

(15 points)

Assume we're working on a machine which has the following parameters:

- 16GiB of physical memory
- 22 bit virtual addresses
- 128B pages
- Each PTE in our single level table is 4B

(a) How many bits are in the Physical Address Offset?

Offset = \_\_\_\_\_ bits

**Solution: 7 bits**

(b) How many bits are in the PPN? the VPN?

VPN = \_\_\_\_\_ bits

**Solution: 15 bits**

PPN = \_\_\_\_\_ bits

**Solution: 27 bits**

helpful! | 0



**Daniel Fan** 1 month ago Physical page size = virtual page size = 128B.

128 =  $2^7$ , so we need 7 offset bits.

virtual address is 22 bits long, so VPN is  $22 - 7 = 15$ .

physical memory is  $2^{34}$ B, which is 34 bits.

PPN is then  $34 - 7 = 27$ .

There's a video walkthrough for sp19 attached in the post above, lmk if there are any questions you have after watching it.

helpful! | 0



**Anonymous Calc 4** 1 month ago thank you!

helpful! | 0



**Anonymous Helix 4** 1 month ago

Hi Daniel. Can I please ask what PTE actually is and what its role? I see its reference even for the next problem, but really can't find useful reference to it when solving the problem. Where would it be used?

helpful! | 0



**Daniel Fan** 1 month ago Hey, I would recommend reviewing this lecture:  
<https://cs61c.org/sp20/lectures/?file=lec19.pdf>. Page 19 describes PTEs.

helpful! | 0



**Anonymous Helix 4** 1 month ago

Yeah I have seen this, but I was not sure what role it plays when determining the number of pages or computing the size of each physical page and etc. Does it play just no role?

helpful! | 0



**Daniel Fan** 1 month ago It's not used for computing the number/size of pages. When you want to translate a VPN to PPN (not just count the bits in each), you index into the page table (which starts at the page table base register) with your VPN and read the corresponding PTE which holds your PPN.

helpful! | 0

Resolved  Unresolved



**Anonymous Atom 4** 1 month ago

[FA-F]:Q9c

Why do we need plus data access time here? And what exactly is that?

**Q9) We've got VM! Where? (15 pts = 2 + 3 + 5 + 5\*1)**

Your system has a 32 TiB virtual address space with a single level page table. Each page is 256 KiB. On average, the probability of a TLB miss is 0.2 and the probability of a page fault is 0.002. The time to access the TLB is 5 cycles and the time to transfer a page to/from disk is 1,000,000 cycles. The physical address space is 4 GiB and it takes 500 cycles to access it. The system has an L1 physically indexed and tagged cache which takes 5 cycles to access and a hit rate of 50%. On a TLB miss, the MMU checks physical memory next.

c) What is the average memory access time (in cycles) for a single memory access for the current process? Assume the page table is resident in DRAM.

**760 cycles**

**SHOW YOUR WORK**

Translation AMAT =  $5 + \frac{1}{2}(500 + 2/1000(1M))$   
=  $5 + \frac{1}{2}(500 + 2000)$   
=  $5 + \frac{1}{2}(2500)$   
=  $5 + 500$   
= 505  
plus  
Data access AMAT =  $5 + 50\% (500)$   
=  $5 + 250$   
= 255  
AMAT (overall) =  $505 + 255 = 760$

helpful! | 0

Resolved  Unresolved



**Anonymous Poet** 1 month ago

Is SP F number 4 in scope? I don't recall amoswap and risc v parallelism...

helpful! | 0



**Anonymous Gear 3** 1 month ago Yes.

helpful! | 0



**Anonymous Gear 3** 1 month ago I think it was in discussion 13 or something like that

Resolved  Unresolved

**Anonymous Gear 3** 1 month ago

[SP-F]:Q6b

This question mentions an optimization of the CPU pipeline with having register reads and writes in the same cycle. Calling write-read, and read-write implementations. Was this something we covered explicitly this semester? Or is it simply a creation of this question?

helpful! | 0

 Resolved  Unresolved

**Anonymous Comp 4** 1 month ago

[SP-F]:Q5c

Can someone explain the notation used along the transition lines? I understand that the states are the registers, but it is not clear what the transition numbers represent.

helpful! | 0

 Resolved  Unresolved

**Anonymous Helix 4** 1 month ago

For problem 3,

May I ask if physical memory = number of pages \* bytes per page? In this case we have PTE (Page table entry) which I thought was the size of each page, but it seems like it's off by a lot, since page size, according to the calculation, seems to be  $2^{27}$  rather than  $2^2=4$ . If they're different, I would really appreciate if someone get shed lights on their difference.

Thanks.

helpful! | 0

 Resolved  Unresolved @367\_f54

**Anonymous Gear 4** 1 month ago

[SP19-F]: Q6b

Can someone explain why variable output is in heap memory instead of on the stack? I thought in a function the variable itself should be stored on the stack.

Also seeing the M1-1 in sp15-final we can find that variable e itself is stored on the stack.

helpful! | 0

 Resolved  Unresolved

**Anonymous Mouse 4** 1 month ago

[FA19-F]:Q5

Can someone please explain why the transition line they gave us is 0/1 instead of 0/0? The two inputs to the or gate right before Q are both directly connected to the output of both Reg1 and Reg2. Since Reg1 and Reg2 both start off with the value 0, shouldn't the OR gate before Q be computing 0 OR 0?

helpful! | 0