# [Midterm] Past Exams - 2021 #635

J Jero Wang STAFF
3 months ago in Exam – Midterm

185
VIEWS

You can find the past exams here: https://cs61c.org/su22/resources/exams/

When posting questions, please reference the semester, exam, and question in this format so it's easier for students and staff to search for similar questions:

**Semester-Exam-Question Number**

For example: **SP21-Final-Q1**, or **SU21-MT2-Q3**

Spring 2021 final walkthrough

Anonymous Cattle 2mth #635bbd ✓ Resolved

**Fall 2021 - Final- Q1**

I'm confused on this question. First of all the answers are determined by knowing the minimum hamming distance between valid code words for a 1bit error is three? So 0x51c and 0x71c are 2 and 1 hamming distances from 0x61c. Thus they can't be a valid codeword. Meanwhile the other options are <= 3 hamming distance and thus could be valid based on our chosen scheme. Is this reasoning correct? Also how do you determine the minimum hamming distance between valid codewords?

...

P Peyrin Kao STAFF 2mth #635bbe

#1300

...

Anonymous Hedgehog 3mth #635bbb ✓ Resolved

**SP21-MT2-Q7-iii**

I am quite confused about what is going on in this 4 lines:

Can you explain especially the first 2 lines (**sll a0 s6 3** & **add a0 a0 s5**)?

Thank you

...

P Peyrin Kao STAFF 3mth #635bbc

#635fa

...

Anonymous Goshawk 3mth #635baa ✓ Resolved

Why is the shift 4, aka a multiplication of 16? The size of a pointer is 4 bytes so shouldnt it be slli t1 t1 2?

...

P Peyrin Kao STAFF 3mth #635bad

That's a typo, it should be 2.

...

Anonymous Reindeer 3mth #635afe ✓ Resolved

**Fall 2021 Midterm**

I understand the opcode and the ra part, but what I don't understand is how we calculated the PC-relative address. We `jal` to `Get20chars` at line 5, so isn't PC supposed to be at line 6? And I am confused about why we need to consider jumping back to the start of the function.

...

A Arda Akman STAFF 3mth #635aff

For this question, the jal jump is happening to the function get20Chars. We know that verifypassword is at address 0x00001000, and the get20Chars is at 0x00000f00, meaning there are 256 bytes in between the addresses of these two functions (256 bytes backward). When working with jal, we need to calculate where the label is in relative to where we were at at the moment (the PC is at the line with jal when the jal command runs).

...

Anonymous Reindeer 3mth #635afc ✓ Resolved

**Fall 2021 Midterm**

Just to clarify, the reason why we are `mv a0, sp` is because `a0` in this case specifically needs 20-byte of space to be able to store the input string right? Normally we don't need to do this because the space for `a0` is enough right? How many bytes does `a0` normally store?

...

P Peyrin Kao STAFF 3mth #635baf

a0 is a register that holds 32 bits = 4 bytes. We now change it to an address on the stack, where we can hold 20 bytes, and use the provided function to write the bytes onto the stack.

...

Anonymous Moose 3mth #635afb ✓ Resolved

Is the exam last summer available?

...

P Peyrin Kao STAFF 3mth #635bae

Probably too late for the midterm, sorry, but we can try to dig it up before the final exam.

...

Anonymous Goshawk 3mth #635adc ✓ Resolved

I'm a little confused, because wouldn't the significand bits be 0101 since those are the first 4 bits after the decimal point? That would indicate to round up.

...

P Peyrin Kao STAFF 3mth #635ade

The exact number is 0b1.01010101... which is closer to 0b1.0101 than 0b1.0110.

...

Anonymous Reindeer 3mth #635ada ✓ Resolved

**Fall 2021**

I don't understand. When you store the string "Hello World" because it is little-endian, shouldn't the lowest address contains "d" iinstead of "H"?

That is how they do it in the lecture too..

...

P Peyrin Kao STAFF 3mth #635adb

Endianness applies to multi-byte values like integers and addresses, not characters. In the slide, the characters are being read from lowest address to highest address.

...

Anonymous Reindeer 3mth #635afd

I see. Because it doesn't really make sense for a 1-byte character to be little-endian right? It has nowhere to go right lol

...

P Peyrin Kao STAFF 3mth #635bba

Replying to Anonymous Reindeer

Yeah, you don't need to put multiple bytes together to form a character.

...

Anonymous Reindeer 3mth #635acf ✓ Resolved

**Fall 2021**

Can someone please explain this? if str2 is not defined in the function but outside the function.

Will this work?

...

P

Peyrin Kao STAFF 3mth #635add

If `char str2[] = "Hello World"` were defined outside the function, the string would be in static memory, so you can return a pointer to the string.

...

V

Victor Shi 3mth #635ace ✓ Resolved

For this implementation:

```
char* foo() {
    char str[] = "Hello";
}
```

the `*str` is in stack because it is created in the function `foo`. However, for this implementation:

```
char str[] = "Hello";
char* foo() {
}
```

Now `str[]` is a global variable, does that mean `*str` is still in the stack? Or in static? Does that mean we cannot change the content of `str` anymore?

...

P

Peyrin Kao STAFF 3mth #635adf

It might help to equivalently write `*str` as `str[0]`, the first character of the string. In the second example, the characters of the string are immutable and in static memory.

...

Anonymous Goshawk 3mth #635acd ✓ Resolved

Why is it true? A 1 byte char, a pointer to uint16_t, a 1 byte char, and another pointer is 10 bytes which, after padding, is 12.

...

P

Peyrin Kao STAFF 3mth #635aea

Answer from a past semester:

basically, since b is a short (uint16_t is 2 bytes), b must start on an address that's divisible by 2, just like how 4-byte things (ints, pointers, etc) must start on an address that's divisible by 4. that's why the extra byte before b is needed.

now say we swap b and c. then the bytes would look like this:

[1 byte for a] [3 bytes of padding to align c] [4 bytes for c] [2 bytes for b] [2 bytes of padding to align d] [4 bytes for d]

we need to pad before c because c needs to start at a 4-byte boundary, and same for d.

...

Anonymous Hornet 3mth #635acc ✓ Resolved

FA21-MT2-Q4.4

This may be a stupid question, but how to transfer from $0b1.11111111 * 2^{15}$ to $2^{16} - 2^{7}$?

Do we need to simplify the $0b1.11111111 * 2^{15}$ to $2^{16} - 2^{7}$ on the test if the question does not mention us to simplify?

**Update**: I'm thinking about $(2^{0} + 2^{-1} + 2^{-2} + ... + 2^{-8}) \times 2^{15}$

...

P Peyrin Kao STAFF 3mth #635aeb

I think it's clearer to start with 0b1 1111 1111 0000 000. This is equivalent to 0b1 1111 1111 1111 111 - 0b1111 111, which is $(2^{16} - 1) - (2^{7} - 1) = 2^{16} - 2^{7}$.

Your update approach works too.

Also, we would accept $2^{7} + 2^{8} + 2^{9} + ... + 2^{15}$, which is equivalent.

...

Anonymous Hedgehog 3mth #635aca ✓ Resolved

**SP21-MT2-Q4.2**

When you say "floating point number", is it always 16 bits?

...

P Peyrin Kao STAFF 3mth #635aec

This is FA21, not SP21. The top of the question defines the floating point system used in this question.

...

Anonymous Squid 3mth #635abf ✓ Resolved

can someone explain mt1 q 6, i dont think I understand the q and ans

...

P Peyrin Kao STAFF 3mth #635aed

Which semester are you asking about? Got a specific question?

...

Anonymous Hedgehog 3mth #635abd ✓ Resolved

**FA21-MT2-Q2**

I want to check what I get wrong about memory chunks,

So **static** is where constants like global variables are stored. So what's there persist throughout the entire life of the program if not explicitly changed.

**Stack** is where things that are temporary are stored, such as local variables. So variables defined in a function frame are stored in stack.

So here, ***str1** is defined inside a function and therefore stored in stack, is what I thought and turned out incorrect. It's actually stored in static.

What part of my logic/knowledge is wrong? Thanks!

...

Anonymous Cattle 3mth #635acb

&str1 is on the stack

str1 is static memory since it points to the address of a string literal.

*str1 is located in static since it's asking essentially where the 'H' is stored which is static since its a string literal.

When youre doing these type of questions just ask yourself what does this evaluate to?

When you see the & operator ask yourself where is my variable being stored (On the stack in this case)? When you see str1 think what it evaluates in this case a memory address thats part of static memory.

...

P Peyrin Kao STAFF 3mth #635aee

`str1`, the 4-byte pointer, is a local variable stored on the stack.

`*str1` , the 12-byte character array actually being pointed at, is a hard-coded string stored in static.

...

Anonymous Tarsier 3mth #635aac ✓ Resolved

So for Q4.3, my answer was 511 because I believe that floating point double counts 0 - there are "positive zero" and "negative zero".

...

P Peyrin Kao STAFF 3mth #635aae

Good catch, I think that would be a valid alternate solution, assuming that you treat +0 and -0 as the same number.

...

Anonymous Chinchilla 3mth #635aab ✓ Resolved

FA21-Midterm-Q1

For question 1.1, the solution says that the preprocessors is part of the compiler. However, doesn't this slide from lecture 10 on CALL indicate that pre-processing is a separate step from compilation?

...

P Peyrin Kao STAFF 3mth #635aad

I think we might have defined it differently this semester, sorry for any confusion. Based on this diagram in this semester's slides, we may have accepted an alternate answer.

...

Charlie Cheng-Jie Ji 3mth #635ff ✓ Resolved

FA21-Midterm-q2

I don't know why 2.2, `*str2` is located on the stack. I thought in lecture we said str2 is located on stack because it's a local variable. Then, how about *str2? (I recall that *str2 is equivalent to str2[0]). Is *str2 also on the stack and why is that?

...

Anonymous Chinchilla 3mth #635aaa

(Maybe a TA could verify this)

My understanding is that array's are created on the stack. So for str2 I think of it as sort of like:

addi sp, sp, -12 #allocate 12 bytes on stack

(# write each char into the allocated space)

and then returning sp + 12, which is the address of the beginning of the array, which is what str2 stores.

So *str2 is 'H' which is stored on the stack.

...

P Peyrin Kao STAFF 3mth #635aaf

This intuition sounds right to me. The brackets in `char s2[]` say that the actual bytes of the array are being stored on the stack (it's not that a pointer is being stored on the stack).

Then, you can note that `*str2` is equivalent to `str2[0]` which gives you the first character of the array.

...

Anonymous Chinchilla 3mth #635abe

Replying to Peyrin Kao

Hi, just to confirm, for the code:
char str2[] = “Hello World”
The characters ‘H’ .... ‘d’, ‘\0’ are stored on the stack, but the local variable str2 is not stored on the stack?
Is this what was meant by “(it’s not that a pointer is being stored on the stack”)

...

P Peyrin Kao STAFF 3mth #635aef

Replying to Anonymous Chinchilla

The local variable `str2` is the actual array, which is stored on the stack.

...

Anonymous Eel 3mth #635fe ✓ Resolved

SP21-Midterm-q8

How were these numbers calculated? The standard exponent bias in this question would be 255 (as there are 9 exponent bits). 0xC07C0000 is 0b1100 0000 0111 1100 0000 0000 0000 0000, so the exponent would be 100000001, which is 513, so 513 - 255 = 258.

...

P Peyrin Kao STAFF 3mth #635abb

0b100000001 is not 513.

...

Anonymous Eel 3mth #635abc

I realize that now, I must have accidentally added a 0 when doing the calculation. Thank you!

...

Anonymous Skunk 3mth #635bab

Where is it stated that there are 9 exponent bits?

...

Anonymous Eel 3mth #635bac

At the top of part B

...

Anonymous Rhinoceros 3mth #635fd ✓ Resolved

sp21-midterm-q4

I'm confused about the circled four lines. Why do we need them? Why can't we just have three lines like `sb t0 0(a0)` `sb t0 1(a0)` and `sb t0 2(a0)`?

...

P Peyrin Kao STAFF 3mth #635aba

Alternate solutions do exist with the approach you suggested. I can't verify them by hand though; I'd suggest running them through Venus if you're curious.

...

Anonymous Eel 3mth #635ec ✓ Resolved

SP21-Midterm-7a-iii

I don't understand how we can iterate through f, if f is a struct. What exactly are we iterating through in the for loop? Also, I don't understand the answer to this question, what is each line of this code doing?

...

P Peyrin Kao STAFF 3mth #635fa

`f` is a pointer to the start of an array of structs (which is a bunch of structs stored side-by-side in memory). Each struct is 8 bytes, so we take the index `i` in s6 (measured in number of structs) and multiply it by 8 to get the offset measured in number of bytes. Then we add the offset to the address of the start of the array `f` in s5. Once we get the address of the struct, we load its `b` field (4 bytes after the start of the struct) into the a0 argument register and call baz.

...

Anonymous Eel 3mth #635fb

How do we know that f is a pointer to an array of structs, instead of just being a pointer to a struct?

...

P Peyrin Kao STAFF 3mth #635fc

Replying to Anonymous Eel

With just the value of the pointer, you have no way of telling whether it points to one struct or multiple structs. You have to rely on the assumptions in the question, or pass in an extra argument telling you the length of the array.

...

Anonymous Rhinoceros 3mth #635dc ✓ Resolved

fa21-mt2-Q5.

I don't understand how we check the calling convention for t register here? Like what's the point? t register is supposed to be saved by the caller. There is just no way we can know if the function modifies t, saves it, or never use it. I can't think of a case where this test will fail due to something related with t register going wrong.

In contrast, the check for s register makes sense, because we are checking that the value is unchanged in the output.

...

R Rohit Agarwal STAFF 3mth #635df

Well there's definitely a point to such a test--you want to make sure if the implementation of `Get20Chars` modifies `t0-t6`, then it doesn't screw up your function!

How you would implement such a test in practice is perhaps use a "mock" version of `Get20Chars`. `mockGet20Chars` would call `Get20Chars` and at the end store random values in `t0-t6`. Then, design your testing framework such that `mockGet20Chars` is called instead of `Get20Chars`. This complexity is why implementation of the test was not required for this question.

...

Anonymous Rhinoceros 3mth #635db ✓ Resolved

fa21-mt2-Q5.

I'm confused here. Isn't sp a register? Why it doesn't have an address?

...

R Rohit Agarwal STAFF 3mth #635dd

Registers can HOLD addresses, but they themselves do not have an address in the memory (RAM). They instead exist on the CPU itself.

...

Anonymous Rhinoceros 3mth #635cf ✓ Resolved

fa21-mt2-Q4.3

Positive and negative infinity is considered to be a number? I thought it would be symbols...

...

R Rohit Agarwal STAFF 3mth #635de

I would consider it a representable "number" or "value". Not sure what you mean by a symbol (numbers are also symbols like 10?)

...

Anonymous Echidna 3mth #635ae ✓ Resolved

sp21-mt2-Q7 b iii

To set the n'th bit to 1, why are we using 1 « (n % 8) instead of 1<<n?

...

P Peyrin Kao STAFF 3mth #635bb

That said, the line you're asking about is described in the comment: "Now we set the n'th BIT in the bloom filter to 1". C indexes in bytes, and 1 byte = 8 bits, so `data[n/8]` is getting the byte where the n'th bit is located.

...

Anonymous Chinchilla 3mth #635cc

However, if we have the data array as something like: 0101 1001 0000 1110 .... and suppose n = 10, we want to update the array to: 0101 1001 0010 1100...
Although data[n/8] selects the correct byte: data[1] = 0000 1110, wouldn't bitwise or-ing that with 1 << n%8 give the wrong result ? Since 1<<10%8 = 1 <<2 = 0100
So 0000 1110 | 0000 0100 wouldn't be the desired result of 0010 1110?
(Basically, my question is that left shifting 1 using n%8 counts the bits from the "right", where as data[n/8] seems to suggest that we are counting the n-th but from the "left"

...

Anonymous Echidna 3mth #635ad ✓ Resolved

sp21-mt2-7b i

I'm confused about how to interpret Line A and get the correct answer. Why do we need data to be the size of "size/8+1"?

...

P Peyrin Kao STAFF 3mth #635bc

`size` is measured in bits, but `calloc` is measured in bytes (1 byte = 8 bits). The +1 lets you round up to the nearest byte. For example, if you need space for 17 bits, 17/8 + 1 = 3 bytes.

...

Anonymous Mole 3mth #635da

why do we use calloc specifically?

...

P Peyrin Kao STAFF 3mth #635ef

Replying to Anonymous Mole

The question says that the bits in the filter are all set to 0 initially, and are set to 1 as the algorithm runs.

...

Anonymous Echidna 3mth #635ac ✓ Resolved

sp21-mt2-Q7 a i

Could you please explain why the sizeof(foo) is 8 but not 5 (1-byte char + 4-byte pointer)? Thank you!

...

A Arda Akman STAFF 3mth #635ba

This is due to padding done by C. More on padding on this thread #216

...

Anonymous Echidna 3mth #635cd

Thanks! So we are using 4-byte alignment here. But I'm still confused about how to choose the number of bytes. Why are we aligning to 4 bytes instead of 8 bytes or 2 bytes?

...

A Arda Akman STAFF 3mth #635ce

Replying to Anonymous Echidna

It depends on the type of value you are using. For example, as the character pointer has a size of 4 bytes, it has to be 4 byte aligned. If we had a 64 bit system , the pointer would be 8 byte aligned and the whole struct would have been 16 bytes instead.

Another thing to consider is that the entire struct will be memory aligned to the size of the biggest value in the struct. So, because the biggest type on the struct foo is 4 bytes, the struct needs to be also 4 bytes aligned (here, that happens automatically).

...

Anonymous Echidna 3mth #635ab ✓ Resolved

sp21-mt2-q2a

"Compilers produce larger code than interpreters but do it faster." why is this correct? I think based on the slides, compilers produce smaller code than interpreters.

...

P Peyrin Kao STAFF 3mth #635ca

I think it honestly depends on the compiler and interpreter you're using, so there isn't one consistent answer across semesters. If we ask this question over summer, the summer lecture slides would be definitive.

...

Anonymous Rhinoceros 3mth #635eb

For the same question, why is this not correct?

...

P Peyrin Kao STAFF 3mth #635ee

Replying to Anonymous Rhinoceros

Depending on your compiler/interpreter, this doesn't need to be true. You could have a really inefficient compiler that produces slow and wasteful code.

...

Anonymous Echidna 3mth #635aa ✓ Resolved

Is sp21-MT2-Q5 in scope?

...

A Arda Akman STAFF 3mth #635af

RISC-V datapath is not in scope

...

Anonymous Gorilla 3mth #635e ✓ Resolved

This is Sp21 Final Q6(a). What are RISC-V alignment rules? The walkthrough video says RISC-V alignment rules mean that each field should be stored starting at multiples of its size. But this alignment rule is neither padding or packing mentioned in lecture. Does RISC-V has different memory alignment rules than C?

...

P Peyrin Kao STAFF 3mth #635cb

I don't think we formally covered RISC-V alignment rules this semester. That said, this is still solvable with the same C padding rules we saw in lecture this semester:

`a` is at byte 0. `b` needs to start at an even address, so we add 1 byte of padding and put `b` at bytes 2-3. `c` takes up bytes 4-7, and `d` takes up bytes 8-11.

...

Anonymous Skunk 3mth #635d ✓ Resolved

In Sp21-MT1-Q3, how does the "Black box" become involved?

...

P Peyrin Kao STAFF 3mth #635bd

The block along the top wire is named the "black box logic." You don't need to know what it does logically, but it has a delay of 9 ns that you use in your calculations.

...

Anonymous Skunk 3mth #635c ✓ Resolved

For Sp21-MT1-Q4, how will we know our code is correct if it doesn't match up directly with the code on the solution? Can we run it in Venus? If so, how?

...

P Peyrin Kao STAFF 3mth #635be

We won't have an autograded question like this on our midterm, since it's in-person and you can't use Venus (even if you're remote). You can use the `print_str` function and hard-code some strings to help test your implementation. To hard-code strings, add something like this at the top of your code:

```
my_string:    .asciiz "some string"
```

and then get the address of the string with a `la` instruction.

...

Anonymous Partridge 3mth #635a ✓ Resolved

Can Someone Explain how to do SP21-MT-Q8 a) and b)? I don't think we covered the step size technique in class, so I'm not to sure how to do part A or part B

...

*This comment was deleted*

Anonymous Partridge 3mth #635f

Oh nice :)

...

P

Peyrin Kao **STAFF** 3mth #635bf

We didn't cover the step size technique officially, but this question is still in scope.

In standard B, there are 45+18=63 bits that can be varied to create different positive numbers (the sign bit has to be 0). When the exponent is all 1s and the significand is non-zero, we have $2^{45} - 1$ NaNs that we have to subtract. The same logic works for standard A.

Step sizes increase as the magnitude of the number increases, so we just need to check the step size around the top part of the range. $80 = 0b1010000 = 0b1.010000 \times 2^6$. The next largest number we can represent is $0b1.010000\ 0000\ 0000\ 0000\ 0001 \times 2^6$ which is $0b0.000000\ 0000\ 0000\ 0000\ 0001 \times 2^6 = 2^{-16}$ larger than 80.

...

Anonymous Vulture 3mth #635ea

When we talk about step size for float A, do we define A's step size as the absolute difference between A and the number just above A? Or do we use the number below A to find the step size?

...

P

Peyrin Kao **STAFF** 3mth #635ed

Replying to Anonymous Vulture

"Step size" isn't an official term, so it's up to the context you use it in. Here, I'm defining step size to be the difference between A and the number just above A.

...