note @434

[Exams] Past Exams 2020 Q&A

Discuss all questions pertaining to exams which took place in 2020 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 of of these: Q, MT, MT1, MT2, F
Please separate out parts with periods: 1.2.ii.a.b.3.a

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

midterm1 midterm2 final

~ An instructor (Jerry Xu) thinks this is a good note ~

Updated 27 days ago by Stephan Kaminsky

followup discussions *for lingering questions and comments*

2 endorsed followup comments

Resolved Unresolved

Anonymous Comp 2 months ago One of the previous midterm 1 papers have questions on the tree data structures. Can we expect things which we haven't done in CS 61C like trees to be a part of the exam?

helpful! | 0

Stephan Kaminsky 2 months ago You will be expected to be able to understand some of the simpler, common algorithms. We generally will not make the algorithms too complex.

helpful! | 0

Resolved Unresolved

Anonymous Comp 2 months ago Ohh okay! But can we expect tree algorithms?

helpful! | 0

Xxxxxxx 2 months ago Tree algorithms were a huge basis in CS 61B (and trees in general were introduced in 61A), so it might be safer to review them via the 61B website or useful websites online.

helpful! | 0

Resolved Unresolved

Anonymous Gear 2 months ago

[SP-MT1]:Q5A

Hey! I was confused on the part of 5A where we bit shifted i by 24; to be honest I didn't understand what the question was asking related to the most important bit.

helpful! | 0

Caroline Liu 2 months ago Essentially what's happening is that the question is saying in an array of ints (4 bytes, 32 bits) the encoded version of the word (array of chars) have each char at the most significant byte (left-most) of each int position in the array. That means each char is 4 bytes away from the next. In order to decide it and not just see ints, we have to cast the bits to char to decode it. However, we most move the bytes to the right-most (least significant) byte (3 over) in order to cast each position to a char (it's taking the array 4 bytes at a type because it was an int array). That's why we need to shift to the right by 3 bytes (24 bits).

helpful! | 1

Resolved Unresolved

Anonymous Scale 2 months ago

[SP-MT1]:Q1C

[SP-MT1]:Q1d

Q1 c)

int doThis = 0x6C697665; -> we need to translate itASCII to "live", right? But do we need to prepare for this?

Q1 d)

Explanation says "Crashes because string literals (“abcd”) are"

but char *boo = “go cardinals!” why this is READ only?

helpful! | 0

Anonymous Poet 2 months ago Hi, for Q1, c, you could check page 15 in lec04 slides;
For Q2, here is the text from Fall 2017 course staff notes. For strings, compilers generally store string literals in the static segment of memory even if the literal is declared inside a function call (and would normally go into the stack). This optimization is made because if a string literal is used, its value will likely not change during the program. However, one exception to this optimization is if a string literal is stored in achar[]instead of achar*. Since an array allocates space for the array's contents, the string literal will be stored in that allocated space in the stack instead of in the static segment.

~ An instructor (Caroline Liu) thinks this is a good comment ~

helpful! | 2

Resolved Unresolved

Anonymous Poet 2 months ago For Q1:a, why sizeof(str) is not correct?
I hold the view that it is because the function sizeof returns unsigned int. However, would the c typecast automatically from unsigned int to int?
If the c could do this, the range of them would be different since one is unsigned and the other is two's complement. So I have no idea whether c could typecast from unsigned int to int.

helpful! | 0

Anonymous Atom 2 months ago [SP-MT1]:Q1.a
In `get_strlen(char* str)`, `return sizeof(str)` would actually be equivalent to `return sizeof(char*)`, so it would always return 4 bytes (in a 32-bit system).

~ An instructor (Caroline Liu) thinks this is a good comment ~

helpful! | 1

Resolved Unresolved

Anonymous Helix 2 months ago
[SP-MT1]:Q2

```
Notice that 8 = 2^3.Thus, we can group 3 binary digits at a time and represen
t them as one octal digit. Since 8 is not a multiple of 3,we can add a zero a
t the beginning of our binary
number as this will not change the value. Now 0b 011 100 011 = 343.
```

Can someone explain why adding a zeor at the beginning will not change the value? I think adding a zero at the beginning of a negative number in 2's complement form will change this value.

helpful! | 0

Anonymous Beaker 2 months ago Since we know we are working with 8 bits, we would cut off the MSB when converting octal back to binary.

helpful! | 0

Resolved Unresolved

Anonymous Mouse 2 months ago [SU-MT1]:Q3c

Can someone explain the process of getting 0xffffff81 from -2? Wouldn't multiplying -2 by -1 yield 2, and right shifting would give 0b00000000000000000000000000000001 = 1 (decimal)? How is the answer -1?

helpful! | 0

Anonymous Beaker 2 months ago The whole problem looks like this:

fun->i[0] = -1 (altering rightmost byte here bc little endian and i contains 8 bit integers)
0000 0000 0000 0000 0000 0000 1111 1111
fun->u[0] *= 2 (altering rightmost byte here bc little endian and u contains 8 bit integers)
0000 0000 0000 0000 0000 0000 1111 1110
fun->t *= -1 (altering all bits bc t is 32 bit integer) ***you're interested in going from this step****
1111 1111 1111 1111 1111 1111 0000 0010
fun->t >>= 1 (altering all bits bc t is 32 bit integer) ***to this step****
1111 1111 1111 1111 1111 1111 1000 0001
fun->s[0] = '\0' (altering rightmost byte here bc little endian and c contains characters which are 8 bits/1 byte)
1111 1111 1111 1111 1111 1111 0000 0000

Hope this helped

helpful! | 5

Resolved Unresolved

Anonymous Calc 2 months ago
[SU-MT1]:Q2a

Is this problem asking for where the pointer(address) is in memory or where the value that the pointer point to in memory? If it is asking where the pointer is in memory, then why every pointer is in the same part as the value it point to ? And how to interpret &receive_buffer is in Static?

helpful! | 0

Jenny 2 months ago The question specifically says " these expressions evaluate to an address value", and we want "region of memory each value corresponds to". If the expression is a pointer variable, the value is the address of what the pointer points to. For &receive_buffer, the expression evaluates to the address of this variable.

helpful! | 0

Resolved Unresolved

Anonymous Comp 2 2 months ago
[SP-MT1]:Q3a

If fun->i[0] = -1, then why does calling fun->u[0] give -1 even though it should contain 0 with calloc?
helpful! | 0

Jenny 2 months ago Remember that for union, it stores different data types in the same memory location and only one member can contain a value at any given time. As we assign fun->i[0] = -1, we changed the value 0 initialized by calloc to -1, so fun->u[0] have value -1 as well.
helpful! | 0

Resolved Unresolved

Xxxxxxx 29 days ago
[SP-F]:Q3

I don't really understand the reading behind any part of this question, except for c. Why is one choice for parallelism better than the other in a, b, and d? Also, I really thought that a and d should have the same answer, why don't they?
helpful! | 0

Anonymous Gear 2 29 days ago Here's how I understand it:
For #1, we have 400Mb data (easily stored in memory) and since we are multiplying by a transposed matrix, we don't have any memory problems so a naive-ish multiplication algorithm should work. However, since you have to keep re-accessing data to compute new entries in the resulting matrix, you will be doing an amount of work linear to the dimensions of the matrices per cell you compute. Thus it would help to have SIMD. Since there doesn't seem to be a mention of additional threads like in #2, I think that sort of rules out MIMD even though this situation would theoretically benefit from MIMD

For #2, we have a lot of data (too much to fit into memory or even the hard disk of normal computers). Therefore we need a cluster of computers to handle all the data which lends itself to Map/Reduce

For #4, we are doing a single operation to compute each cell value. Since we don't revisit cells like we do in the multiplication algorithm, we can say that it is memory bound since we don't have the benefits of looking back in cache a lot (due to the fact we don't re-use any info from cells). Therefore the benefits of SIMD are diminished since the bottleneck will be the work done to load new locations in memory. Again, there seems to be no mention of additional threads so I guess that sort of rules out MIMD (see explanation to #1)
helpful! | 3

Resolved Unresolved

Anonymous Gear 2 29 days ago
[SP-F] Q7 (c) v.
The prompt says to divide by size 8 with a shift, which I would assume to be a right shift since this is what the C code above says. However the solution key says to do srli which is a left shift. Am I misunderstanding something here?
helpful! | 1

Xxxxxxx 28 days ago I think srli is a right shift... the 'l' stands for logical, not left.
helpful! | 1

Anonymous Gear 2 28 days ago I read it as slli oops

helpful! | 0

Resolved Unresolved

Anonymous Mouse 29 days ago [SU-F] Q1.13

Is there always a chance that a data race can result in the correct output? I was under the impression that data races always caused a data error.

helpful! | 0

Anonymous Scale 2 28 days ago Not sure, but my guess is that it depends on the program. In this case, we don't know anything about the order of computation or when threads are active. It's possible that we end up with no two threads working simultaneously throughout the execution.
Also, consider the case where we run the program on a single core, single threaded CPU. OMP will distribute the work among the one (1) thread.

helpful! | 0

Resolved Unresolved

Anonymous Mouse 29 days ago [SU-F] Q3

Is there an additional handout that was not included in the PDF that we are supposed to use to help us with these questions?

helpful! | 0

Anonymous Gear 2 27 days ago @433

helpful! | 0

Resolved Unresolved

Xxxxxxx 28 days ago

[SP-F]:Q4.b

I'm not sure how I should go about solving this one. On the solution it says they figured out the VPN of 0x00111999 was 0x00111, and I'm not sure where that came from. When I did it, I converted 0x00111999 into decimal, then divided it by 4000 (because they is the number of bytes in a page) and got a VPN of 0x00118 instead.

helpful! | 0

Anonymous Gear 2 28 days ago If you calculate the size of the VPN, you should get 20 bits. Since each virtual address is in the format [VPN][Offset], we take the top 20 bits (aka 5 hex values) which is 0x00111 in this case. Remember that we find our table entry by indexing via the VPN. We use the PTBR to find the start of table and then index from there. You can think of it almost like a map in that you have VPN 0x00000 pointing to the same address as the PTBR, VPN 0x00001 pointing to the next entry, and so on. In order to actually find the physical address of where we are storing the entry, we need to multiply by the size of each PTE as this is sort of a scaling factor. If we were to actually try and access 0x20000000 + 0x00001 then we'd be accessing the middle of a PTE which is not what we want, we want to be looking only at the start addresses of these PTEs. Since our systems are byte addressed and each PTE is 2 bytes in this instance, we multiply by 2 to get 0x20000000 + 0x00222

helpful! | 1

Resolved Unresolved

Anonymous Poet 2 28 days ago
[SP-F]: Q2.b

Why is the number of hits 0?
helpful! | 0

Xxxxxxx 28 days ago Because the cache is only big enough to store half of the array. We load the first N/2 items, then once we load item number (N/2 + 1) it replaces item 1, and each successive item replaces what we already have in the cache until we're left with the cache holding the second half of the array. Then we repeat all that over and over 30 times.
helpful! | 0

Anonymous Poet 2 28 days ago Yeah that's what I got too. I just wanted to make sure my reasoning was right. Thanks for responding!
helpful! | 0

Anonymous Helix 2 27 days ago when we load the first N/2 items, let's say N=5; when i=1 (2nd inner loop iteration), don't we have a hit since we just loaded first N/2 iterms?
helpful! | 0

Anonymous Helix 2 27 days ago nvm just realized i was inner loop not outer
helpful! | 0

Resolved Unresolved

Anonymous Atom 2 28 days ago
[SP-F]: Q2.a.iii

Why isn't it taking advantage of spatial locality as well? I thought since a[i] is right next to a[i-1] and a[i+1] and the loop is accessing them one by one, it is also exploiting spatial locality.
helpful! | 0

Anonymous Gear 2 27 days ago Since the block size is 4 bytes, you are loading in one element of the array into each block. The way I understand spatial locality is that, let's say you have block size = 8 bytes and you load in the first element of the array. When you go to access the second element, you have a hit because loading in the first element brought in the second element into the same block. Here you don't really have that kind of behavior since we have N compulsory misses on the first iteration and then we use temporal locality (i.e. accessing a memory location that was recently put into cache) by iterating over the same array
helpful! | 0

Resolved Unresolved

Anonymous Atom 2 28 days ago
[SP-F]: Q3

Why is part c MIMD? if all those threads communicate through queues, they follow a FIFO order, and results wouldn't be correct if the order is messed up?

For part d, why is the speedup minor?
helpful! | 0

Anonymous Gear 2 27 days ago @434_f10

helpful! | 0

Resolved Unresolved

Anonymous Atom 2 28 days ago
[SP-F]Q5c.iii:

Since CF and CB, and Mem are all 1 bit, is it ok if we used ~ instead of !
helpful! | 0

Anonymous Gear 2 27 days ago It should technically be ok but if we are representing these CF/CB/Mem variables with regular ints then that would cause problems
helpful! | 0

Resolved Unresolved

Xxxxxxx 27 days ago
[SP-F]Q5b.iv:

Can someone explain why the answer is Heap Address? I'm confused about double/single pointer dereferencing.
helpful! | 0

Max Litster 27 days ago u is a pointer to the heap, which we get from the call to malloc.

v is defined as the address of u, or equivalently, a pointer to u. Therefore v is a pointer to an int64_t pointer, hence the syntax int_64t ** v.

When I dereference a double pointer, it is the same thing as dereferencing a normal pointer, only instead of getting back something like a char or an int, I get a pointer.

Thus, when I dereference v, I get u, a pointer to the heap, or a heap address.
helpful! | 1

Resolved Unresolved

Anonymous Atom 2 27 days ago
[SU-MT1]Q1.a.i:

I thought the PTE needs to address all of virtual memory, and not physical memory? After all it's programs tryna access the PTE and from their view they can only see VM.

(a) Consider a system with 4 GiB of physical memory and 64 GiB of Virtual Memory. The page size is 4 KiB. Recall that the page table is stored in physical memory and consists of PTE's, or page table entries. Please fully simplify your answer and leave it in decimal. Fully simplify your exponents down to decimal! Please round your decimal values to two places if needed (do not include unnecessary 0's).

i. (3.0 pt) If, for each PTE, we choose to also store 12 bits of metadata (e.g. permission bits, dirty bit), how many page table entries can we now store on a page?

1024

First we need to find the side of each PTE so we need to figure out how many bits of physical memory we need to have to address. $\log_2(\frac{4GiB}{4KiB}) = \log_2(\frac{2^{32}}{2^{12}}) = 20bits$

Now we can calculate the size of each PTE: 20bits (number of physical pages)+12bits (metadata bits) = $32bits = 4Bytes$

Then we need to find the size of a page: $4KiB = 4096Bytes$

Then we divide: $4096Bytes/4Bytes = 1024$

helpful! | 0

Anonymous Gear 2 27 days ago Each PTE only contains the PPN. The number of bits in the PPN is 20 bits as shown by the calculation there. You need log_2(number of physical pages) bits for your PPN. Now you add on the number of metadata bits for your PPN and get 32 bits = 4 bytes. Since each page is 4096 bytes, you divide by 4 bytes per PTE and get 1024

helpful! | 0

Resolved Unresolved

Xxxxxxx 27 days ago

[SU-MT1]Q1a.b:

Can someone clarify this question:

How many pages does our page table occupy (aka how many valid (active) pages is ourpage table) if we have a two level page table which has only one valid data page? Each level uses an equal amount of bits of the page number.

What does "each level uses an equal amount of bits of the page number" mean? I'm having trouble understanding how to connect "using bits" to page tables. How do we know each level has 12 bits?

helpful! | 0