CS3330 Practice Exam 3 – Fall 2014

Name: _____

Directions: Put the letter of your selection or the short answer requested in the box. Longer answers may use the space to the left of the box too. Write clearly: if we are unsure what you wrote you will get a zero on that problem.

The final exam will also include questions similar to those on the first two exams.

. . . . . .

Question 1: You are writing some code that computes the color of each pixel in a real-time 3D game, which is the most time-consuming part of the game. Rank the following in order of priority

A find exactly the color the artist requested
B find approximately the right color
C find the color quickly
D make the code readable

Answer:

Question 2: What is the main reason to inline procedure calls during optimization?

A because procedure calls exhibit poor spatial locality
B because procedure calls exhibit poor temporal locality
C because procedure calls make it hard for the compiler to optimize your code
D because procedure calls involve a lot of time-consuming stack manipulation
E because procedure calls usually involve branch mispredictions
F because procedure calls hide what's really going on from you, the optimizing programmer

Answer:

Question 3: In a fault, the exception number comes from

A a register value set before the fault occurred
B the system bus
C the parameter of the opcode that caused the fault
D the exception that caused the fault
E the opcode that caused the fault

Answer:

Question 4:
Draw the buses and devices that connect the CPU, memory, the Disk, and at least one other peripheral (network card, graphics adapter, etc).

Question 5: Multiple accumulators are used to take advantage of what modern hardware characteristic?

A math in the ALU is pipelined
B there are only a few registers available
C memory reads can take several cycles
D there are several cores available

Answer:

Question 6: Big-O expresses the asymptotic behavior of code, ignoring all multiplicative constants and low-order terms. In optimization we often discuss ideas like Cycles per Element (CPE), which express

A just the multiplicative constant on the high-order term
B the low-order terms
C the full time equation, constants and low-order terms included
D the high-order term with its multiplicative constant, but not the low-order terms

Answer:

Question 7: Exceptions are handled by code that the hardware locates from

A a special register plus the exception number
B a special register
C the exception number
D none of the above

Answer:

Question 8: Since the OS stores information about each allocated page individually (e.g., where it is stored in physical memory and on disk, if it is read-only, if it is executable, etc), why does it also have a `vm_area_struct`? Select all that apply.

A it allows smaller-than-page-level control of memory
B "also" is misleading; the `vm_area_struct` is (part of) what stores information about a single page
C it makes the `.bss` section (uninitialized global data) more efficient
D it lets the operating system exercise more control than the (hardware controlled) page table entries allow

Answer:

Question 9: In an interrupt, the exception number comes from

A the system bus
B the parameter of the opcode that caused the interrupt
C the opcode that caused the interrupt
D a register value set before the interrupt occurred
E the exception that caused the interrupt

Answer:

Question 10: In a Linux system call, the action the OS is to take comes from

A the system bus
B a register value set before the interrupt occurred
C the opcode that caused the interrupt
D the parameter of the opcode that caused the interrupt
E the exception that caused the interrupt

Answer:

Question 11: Which of the following are reasons to use virtual addresses? Select all that apply.

A let every process use the full address space
B detect null pointer dereferences
C let code ignore the amount of RAM actually available
D makes memory allocators ('malloc', 'new', etc) easier to write
E allow input devices to write to memory while the processor works on something else
F makes it harder for a broken program to mess up other programs
G simplify sharing libraries between many processes

Answer:

Question 12:
Fix the memory error in the following code:

```
int **A = (int**)malloc(n * sizeof(int))
```

Question 13:
If your computer has R bytes of RAM, E-byte page table entries, P-byte pages, and L levels in the page table hierarchy, write an equation for the number of bits in the virtual address space. Use lower case letters as the $\log_2$ of their capital equivalents (e.g., $e = \log_2(E)$). If you need to know things other than R, E, P, and L, write "need info" as your answer.

Answer:

Question 14: Which of the following makes it hard for a compiler to optimize your code? Select all that apply

A using procedure
B pointer parameters
C using `structs` instead of primitive types
D using lots of files

Answer:

Question 15: Loop unrolling reduces which of the following sources of inefficiency? Select all that apply

A loop counter increments
B jumps
C checks of the guard expression
D procedure invocations

Answer:

Question 16: The code `int f() { return malloc(sizeof(int))[0]; }` exhibits which of the following errors? Select all that apply.

A memory leak
B dereferencing a bad pointer
C potential for buffer overflow
D reference to nonexistent variable
E reading uninitialized memory
F dereferencing freed memory
G off-by-one error

Answer:

Question 17:
Draw a pipeline diagram for the following code, assuming there is no data forwarding:

```
irmovl $10,%edx
irmovl  $3,%edx
addl  %edx,%eax
halt
```

Question 18: In a trap, the exception number comes from

A the exception that caused the trap
B the system bus
C the opcode that caused the trap
D the parameter of the opcode that caused the trap
E a register value set before the trap occurred

Answer:

Question 19: We have removable loop inefficiencies when

A the loop iterates more times than it needs to
B the loop repeats computations which give the same answer every time
C the loop spends too much time on looping overhead because its body is small
D the loop contains more memory references than it needs

Answer:

Question 20: Adding local variables adds speed under which of the following circumstances?

A if the expressions they are replacing are memory references
B if the expressions they are replacing are global variables
C if the expressions they are replacing involve arithmetic
D if the expressions they are replacing were defined a few hundred lines earlier in the same method

Answer:

Question 21: Which of the following is unchanged when mapping from virtual to physical addresses?

A some (but not all) page number(s)
B page offset
C TLB tag
D TLB index
E all page numbers

Answer:

Question 22:
Re-write the following expression in a more optimal form:

```
a[0] + a[1] + a[2] + a[3]
```

Question 23: Looking at the Core i7, we notice that the L1 cache index and cache offset together fit inside the physical page offset. Why is this?

A it's so we can begin fetching from L1 before address translation finishes
B it just coincidence, not design
C it's so that we have the information needed for the L2 cache early
D none of the above

Answer:

Question 24:
If your computer has R bytes of RAM, E-byte page table entries, P-byte pages, and L levels in the page table hierarchy, write an equation for the number of bits in the physical address space. Use lower case letters as the $\log_2$ of their capital equivalents (e.g., $e = \log_2(E)$). If you need to know things other than R, E, P, and L, write "need info" as your answer.

Answer:

. . . . . .