CS 61C Midterm #2 — November 13, 2003

Your name ________________________________

login cs61c–_______

Discussion section number _______

TA's name ________________________________

This exam is worth 30 points, or 15% of your total course grade. The exam contains thirteen substantive questions, plus the following:

Question 0 (1 point): Fill out this front page correctly, including the signature, and put your name and login correctly at the top of each of the following pages.

This booklet contains six numbered pages including the cover page. Put all answers on these pages, please; don't hand in stray pieces of paper. This is an open book, open notes exam; no electronic devices allowed.

READ AND SIGN THIS:

I certify that my answers to this exam are all my own work, and that I have not discussed the exam questions or answers with anyone prior to taking this exam.

If I am taking this exam early, I certify that I shall not discuss the exam questions or answers with anyone until after the scheduled exam time.

____

0	/1
1–3	/5
4–6	/6
7	/6
8–10	/6
11–13	/6
total	/30

Question 1 (1 point):

Which part of the translation process is responsible for register `$1` (`$at`)? (Choose one answer.)

______Loader

______Linker

______Assembler

______Compiler

Question 2 (2 points):

Which of the following is *not* a job of the linker? (Choose one answer.)

______relocation

______compute branch offsets

______combine .o files

______resolve external symbols

Question 3 (2 points):

You have a choice between two disks which are identical except for seek time and transfer rate:

A. 5 ms avg. seek time, 50 MB/s transfer rate

B. 10 ms avg. seek time, 100 MB/s transfer rate

Which would you choose based upon the following access patterns? (Choose either A or B for each pattern.)

Small 1K files: ______

Large (1MB) files stored in contiguous blocks on disk: ______

Your name ____ login cs61c–_______

Question 4 (2 points):

For each of the following protocols, indicate its level in the protocol stack by writing one of the words Link, Network, Transport, or Application:

TCP ____________

HTTP ____________

IP ____________

Ethernet ____________

UDP ____________

Question 5 (2 points):

Imagine that two networked computers, A and B, have a one-way network latency of 2ms and a network bandwidth of 1 Mbit/sec (10^6 bits/sec). How long, in milliseconds, does it take between the time A starts sending a 125 byte (total, including headers, etc.) data packet to B, and the time A receives the entire acknowledgement (ACK) packet? Assume ACKs are the same size as the data packet, and it takes a computer 1ms to generate an ACK in response to receiving an entire incoming packet.

Question 6 (2 points):

Why can't interrupt handlers use the stack pointer they find in `$sp`? Choose the best answer.

______The interrupt handler is in kernel memory while the stack resides in user memory.

______An interrupt might occur while the stack pointer is being updated.

______The interrupt handler might overflow the application's stack space.

______Interrupt handlers have no need for a stack.

Question 7 (6 points):

While working on project 4 (the interrupt project), your friend asks you to to help debug the receiver interrupt portion of his/her code. Explain what is wrong, **either by changing the code or by writing a single English sentence and showing where the problem is in the existing code,** for each bug. The bugs do not involve enabling and disabling of interrupts, saving and restoring the EPC, or any part of the kdata section. **There are three bugs.**

```
            .ktext 0x80000080

intrp:      mfc0  $k0, $14
            addiu $sp, $sp, -24
            sw    $t0, 0($sp)
            sw    $t1, 4($sp)
            sw    $t2, 8($sp)
            sw    $t3, 12($sp)
            sw    $t4, 16($sp)
            sw    $k0, 20($sp)

            lui   $t0, 0xffff          # Everything above this line is correct
            lw    $t1, 0($t0)
            beq   $t1, $0, xmt_intrp   # Check if the receiver is ready

            lw    $t2, rec_nextIn
            lw    $t3, rec_nextOut
            addiu $t2, $t2, 1          # Check if the input
            beq   $t2, $t3, xmt_intrp  # buffer is full

            lw    $t4, 4($t0)          # Load from the rcv data register
            lw    $t2, rec_nextIn
            sb    $t4, rec_buffer($t2) # Store into the input buffer

            addiu $t2, $t2, 1
            sw    $t2, rec_nextIn      # Update nextIn pointer

xmt_intrp:  ...                        # Everything below this line is correct

            lw    $k0, 20($sp)
            lw    $t4, 16($sp)
            lw    $t3, 12($sp)
            lw    $t2, 8($sp)
            lw    $t1, 4($sp)
            lw    $t0, 0($sp)
            addiu $sp, $sp, 24
            rfe
            jr    $k0

            .kdata

rec_buffer: .space 8

rec_nextIn:  .word 0
rec_nextOut: .word 0
```

Your name ____ login cs61c–_______

Question 8 (2 points):

A cache access takes 30ns and a memory access takes 200ns. The cache is accessed first; if the desired address is not found in the cache, then main memory is accessed. What's the minimal cache hit rate necessary for the cache to improve average performance?

Question 9 (2 points):

Caculate the sizes (number of bits) of the fields for the given caches.

(a) The cache's total data capacity is 64 Kbytes, the block size is 32 words, and the cache is direct mapped.

tag ____________

index ____________

offset ____________

(b) The same cache as in (a), but 4-way set associative.

tag ____________

index ____________

offset ____________

Question 10 (2 points):

(a) As you go from direct mapped to fully associative, your hit *rate* will usually (pick one):

______decrease ______increase ______not change

(b) As you go from direct mapped to fully associative, your hit *cost* (time required) will (pick one):

______decrease ______increase ______not change

Question 11 (2 points):

A machine has 32 bit virtual and physical addresses. The page size is 16K bytes. Here is the TLB:

```
  Virtual Page Number  Physical Page Number
  +-------------------+--------------------+
0 |        0x0789     |       0x0030       |
1 |        0x0084     |       0x050a       |
2 |        0x0021     |       0x1000       |
3 |        0x0800     |       0x5683       |
  +-------------------+--------------------+
```

What physical address corresponds to the virtual address `0x002124f8`?

Question 12 (2 points):

Here are four concepts relating to virtual memory:

A. working set
B. TLB
C. multitasking
D. LRU replacement

It can be sensible for a computer's virtual address space to be bigger than its maximum possible physical memory size because of (choose one of A–D) ____________.

It can be sensible for a computer's virtual address space to be smaller than its maximum possible physical memory size because of (choose one of A–D) ____________.

Question 13 (2 points):

Here are eight concepts relating to virtual memory:

A. working set
B. write-through
C. write-back
D. LRU replacement
E. random replacement
F. shared code
G. heap allocation
H. set associativity

For each of the four page table entry flag bits, indicate which concept is most relevant (choose one of A–H):

Valid ____________ Referenced ____________

Writeable ____________ Dirty ____________