Computer Science 61C
Midterm 2 - Fall 1995

Professor Harvey, B

Your Name____________________________________

login cs61c-___

Discussion section number ____

TA's name ____________________________________

This exam is worth 25 points, or 18.7% of your total course grade. The exam contains eight substantive questions, plus the following:

Question 0 (1 point): Fill out this front page correctly 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 exam.

Page 1

Question 1 (2 points):

You are compiling the following C procedure to MAL:

```
void foo(char a, char b) {
    baz(a); baz(b); baz(c);
}
```

Recognizing that your need to save the arguemnts on the stack, you begin your translation this way:

```
foo:    addi $sp, $sp, -6
        sb   $4, 0($sp)
        sb   $5, 1($sp)
        sw   $31, 2($sp)
        ...
```

In one sentence, what will go wrong when you run this procedure?

Question 2 (2 points):

Here are two C procedures. Suppose you are compiling them to MAL. You would like to *avoid* using a stack frame if possible. For each procedure, decide whether or not it can be compiled *without* using a stack frame, and circle it if so.

```
int proc1(int x, int y, int z) {
    int a, b;

    return (a*x)+(b*y)+z;
}


int proc2(int x) {
    return garply(x)+3;
}
```

Page 2

Question 3 (2 points):

Here are two C procedures. Each of them may or may not have a bug. Circle the incorrect procedure(s) and explain the bug(s) **in one sentence each.**

```
char *five(char ch) {
    /* returns a string of five copies of the argument char */
    char str[6];
    int i;

    for (i=0; i
```

Page 3

Question 4 (3 points):

You have a direct mapped cache with 16K data words (64K bytes), arranged in blocks that are four words (16 bytes) wide. You run the following program:

```
int array[SIZE], i;

/* initialization */

for (i=0; i &lt SIZE; i += STRIDE) array[i] = 0;

/* second pass */

for (i=0; i &lt SIZE; i += STRIDE) array[i] = 1;
```

During the second pass only, for each of the following sizes and strides, what cahce hit rate would you expect:

(a) SIZE = 64K words, STRIDE = 16K words.

_____Nearly 100% hit rate.

_____About 50% hit rate

_____Near zero hit rate

(b) SIZE = 16K words, STRIDE = 2 words.

_____Nearly 100% hit rate.

_____About 50% hit rate

_____Near zero hit rate

(c) SIZE = 32K words, STRIDE = 2 words.

_____Nearly 100% hit rate.

_____About 50% hit rate

_____Near zero hit rate

Page 4

Question 5 (3 points):

Given a cache size of 64 bytes (yes, that's small!) with a width of two words (8 bytes), and four-way set associative access, with the following contents (all numbers in hexadecimal):

```
set  tag        word 0      word 1

0    1232123    12345678    abcd4321
     0123212    54545454    98701356
     048c848    9999ffee    05551212
     0919091    7111248d    fab00123
1    1232123    12345678    abcd4321
     0123212    54545454    98701356
     048c848    9999ffee    05551212
     0919091    7111248d    fab00123
```

Circle the byte in the cache corresponding to address 0x12321239, supposing that this machine is little-endian, so that the lowest-address byte of a word is the rightmost byte.

Question 6 (3 points):

Under what circumstances might a valid page table entry have its Modfied bit on, but its Referenced bit off?

Question 7 (3 points):

True of false: Becaues the penalty for a page fault is so great, the replacement policy for TLB slots must be LRU rather than Random. Explain your answer **in one sentence.**

Page 5

Question 8 (6 points):

Translate the follwing C procedure to MAL:

```
int weird(int flag, int *arg) {
    int value;

    if (flag <0)
        value = firstFunc(arg[0]);
    else
        value = secondFunc(arg[0]);
    return (value + arg[1]);
}
```

Page 6