

Practice Problems for Topic 8

CIS*2030: Structure and Application of Microcomputers

The practice problems below are important, but will *not* be marked. Their purpose is to ensure that you understand the major concepts covered in Topic 8. Doing these problems by yourself is imperative, as a portion of the marks on the final exam will be based on questions related to Topic 8.

1. A partial C function has the following overall structure:

```
void fun(int a, int *b)
{
    -----
}
```

Recall that C pushes parameters onto the stack from right to left. Below, a C compiler for the 68000 generates the following assembly code for the previous function:

```
_fun LINK      A6, #0
      MOVEA.L  10(A6), A4
      MOVE     (A4), D1
      MULS     8(A6), D1
      MOVE     D1, (A4)
      UNLK     A6
      RTS
```

- a. Draw a memory map showing where parameters a and b as well as the previous frame pointer are located on the stack immediately after the LINK instruction executes. Remember to show the location of the stack pointer and the new frame pointer, as well as the displacements of all items on the stack relative to the new frame pointer.
- b. With the aid of the previous memory map, reverse engineer the assembler code and fill in the three missing parts of the original C code indicated by the dotted line in the C code.

This document is IP.

Please do not share or post on a website.

2. A partial C function has the following overall structure:

```
void fun(int x, int y)
{
    int val = -----;
    if(-----) {
        if(-----) {
            val = -----;
        }
        else
            val = -----;
    } else if (-----)
        val = -----;
    return val;
}
```

Below, is the assembler code generated by a C compiler for the complete function:

```
_fun LINK A6,#-2
    MOVE.W    8(A6),D0
    ASL.W     #2,D0
    MOVE.W    D0,-2(A6)
    TST.W     10(A6)
    BLE       L1
    MOVE.W    8(A6),D0
    CMP.W     10(A6),D0
    BGE       L2
    SUB.W     10(A6),D0
    MOVE.W    D0,-2(A6)
    BRA       L3
L2    AND      10(A6),D0
    MOVE.W    D0,-2(A6)
    BRA       L3
L1    CMPI.W   #-2,8(A6)
    BGE       L3
    MOVE.W    8(A6),D0
    ADD.W     10(A6),D0
    MOVE.W    D0,-2(A6)
    UNLK      A6
L3    RTS
```

- a. Draw a memory map showing where parameters a and b as well as the previous frame pointer are located on the stack immediately after the LINK instruction executes. Remember to show the location of the stack pointer and

the new frame pointer, as well as the displacements of all items on the stack relative to the new frame pointer.

- b. With the aid of the previous memory map, reverse engineer the assembler code and fill in the three missing parts of the original C code indicated by the dotted line in the C code.