Additional Examples (on String)

Sparsh Mittal

Determining string length

- C-Implementation:

```
int strlen(const char *str) {
    int i;
    for (i = 0;str[i] != '\0';i++);
    return i;
}
```

https://marz.utk.edu/my-courses/cosc230/book/example-risc-v-assembly-programs/

Determining string length (RISC-V).

```
        .section .text
        .global strlen

strlen:
                                # a0 = const char *str
        add  t0, zero, zero     # i = 0
start:                          # Start of for loop
        add  t1, t0, a0         # Add the byte offset for str[i]
        lb   t1, 0(t1)          # Dereference str[i]
        beq  t1, zero, stop     # if str[i] == 0, break for loop
        addi t0, t0, 1          # Add 1 to our iterator
        jal  zero, start        # Jump back to condition start
stop:                           # End of for loop
        addi a0, t0, 0          # Move t0 into a0 to return
        jalr zero, ra           # Return back via the return address register
```

https://marz.utk.edu/my-courses/cosc230/book/example-risc-v-assembly-programs/

String Copy Example: C-code

- Remember: in C, strings are null-terminated

```
    void strcpy (char x[], char y[])
{ size_t i=0;
  while ((x[i]=y[i])!='\0')
    i += 1;
}
```

- Base addresses for arrays x and y are in x10 and x11
- i is in x19.

String Copy Example:RISC-V

```
     strcpy:
     addi sp,sp,-8                # adjust stack for 1 doubleword
     sd   x19,0(sp)               # push x19
     add  x19,x0,x0          # i=0
L1: add  x5,x19,x11          # x5 = addr of y[i]
     lbu  x6,0(x5)                 # x6 = y[i]
     add  x7,x19,x10         # x7 = addr of x[i]
     sb   x6,0(x7)                 # x[i] = y[i]
     beq  x6,x0,L2                 # if y[i] == 0 then exit
     addi x19,x19,1          # i = i + 1
     jal  x0,L1              # next iteration of loop
L2: ld   x19,0(sp)           # restore saved x19
     addi sp,sp,8            # pop 1 doubleword from stack
     jalr x0,0(x1)                 # and return
```

Reverse a string (C-Implementation).

```
void strrev(char *str) {
    int i;
    int sz = strlen(str);
    for (i = 0;i < sz / 2;i++) {
        char c = str[i];
        str[i] = str[sz - i - 1];
        str[sz - i - 1] = c;
    }
}
```

https://marz.utk.edu/my-courses/cosc230/book/example-risc-v-assembly-programs/

Reverse a string (RISC-V)

```
            .section .text
            .global strrev

strrev:
                                # s1 = str
                                # a0 = sz
                                # t0 = sz / 2
                                # t1 = i

                                # Enter stack frame
    addi    sp, sp, -16
    sd      ra, 0(sp)
    sd      s1, 8(sp)

                                # Get the size of the string
    mv      s1, a0
    call    strlen
    srai    t0, a0, 1           # Divide sz by 2
    li      t1, 0               # i = 0
start:                          # for loop
    bge     t1, t0, stop
    add     t2, s1, t1          # str + i
    sub     t3, a0, t1          # sz - i
    addi    t3, t3, -1          # sz - i - 1
    add     t3, t3, s1          # str + sz - i - 1
    lb      t4, 0(t2)           # str[i]
    lb      t5, 0(t3)           # str[sz - i - 1]
    sb      t4, 0(t3)           # swap
    sb      t5, 0(t2)
    addi    t1, t1, 1
    j       start
stop:
                                # Leave stack frame
    ld      s1, 8(sp)
    ld      ra, 0(sp)
    addi    sp, sp, 16
    ret
```