CS 315-02 RISC-V Assembly 1

Project 01 Interactive Grading

Project 01 Q & A

Typedef

uint32_t

$$\text{typedef } \underbrace{\text{unsigned int}}\ \underbrace{\text{uint32\_t}};$$

```
struct config_st {
        int count;
        bool header;
        bool footer;
})
```

```
struct config_st config;
```

```
typedef struct config_st config_t;

config_t config;
```

```
typedef struct {
        int count;
        bool header;
        bool footer;
} config_st;

config_st config;
```

```
typedef struct config_st *config_p;

int foo(config_p cp) {
        cp->count = 2;
}
```

separate compilation

```
gcc -o numconv numconv.c numhelpers.c
```

numinfac → main()

→ funcs
numhelpers.c

numhelpers.h
prototypes

numconv.c

```
#include <stdio.h>
   ⋮

#include "numhelpers.h"
```

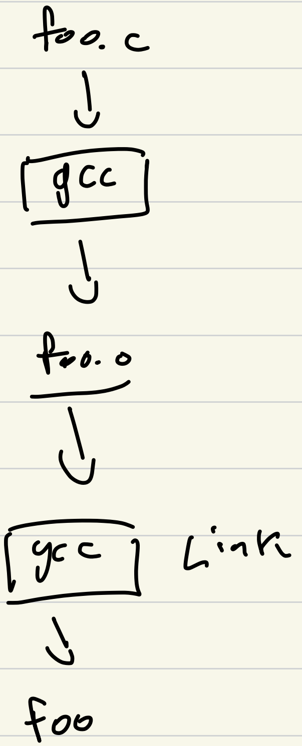
Makefile

```
NUMCONV_OBJS = numconv.o numhelpers.o
```

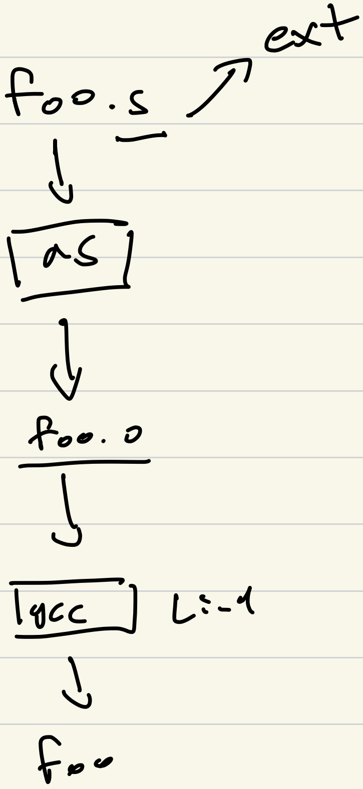
RISC-V Assembly Language

Assembly Language → Human readable form of machine code
machine language

Compiling



Assembling



Vocabulary

$$\overbrace{\underbrace{\text{add}}_{\text{mneumonic}}}^{\text{instruction name}}\ \ \overbrace{\underset{\substack{\uparrow\\ \text{dest}\\ \text{reg}}}{a0},\ \underbrace{a1,\ a2}_{\substack{\text{source}\\ \text{operands}\\ \text{(regs)}}}}^{\text{operands}} \qquad a0 = a1 + a2$$

$$\left.\begin{matrix}a0\\ a1\\ a2\end{matrix}\right] \text{— registers}$$

processor
variables

registers : (on RISC-V 64 bit)

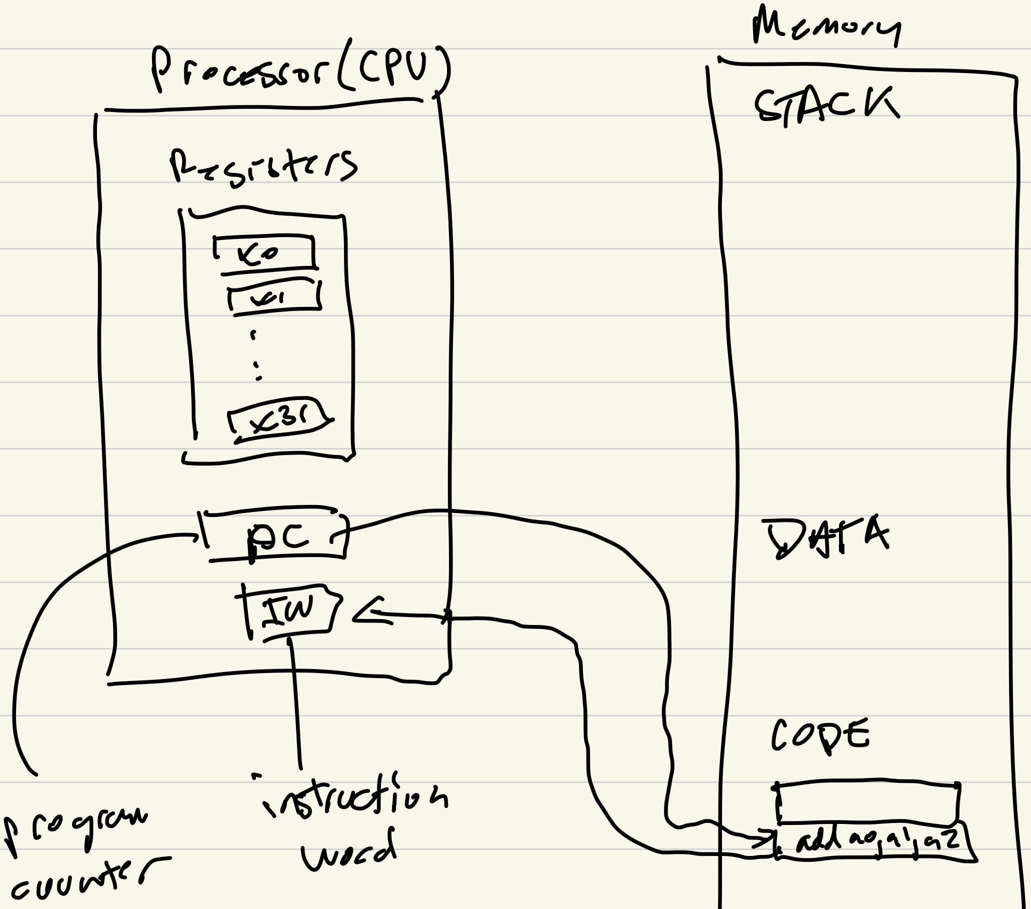
32 regs, each reg is 64 bits

registers: x0, x1, x2, ..., x31

a0, a1, a2, ... arguments

t0, t1, t2, ... temporary value

Machine Code Execution Model



myInt

my_int

Assembly Source Components

labels
instructions
directives
comments

RISC-V Assembly Types of Instructions

3 catagories

1) Data processing — add
sub
mul

2) Control — ret

3) Memory