CS 315-01 RISC-V Assembly 1

Project 01 Q & A

Project 01 Interactive Grading

typedef

uint32_t

```
typedef unsigned int 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 headec;
    bool footec;
} config_st;

config_st config;
```

```
typedef struct config_st config_t;

int foo ( config_t c ) {


}
```

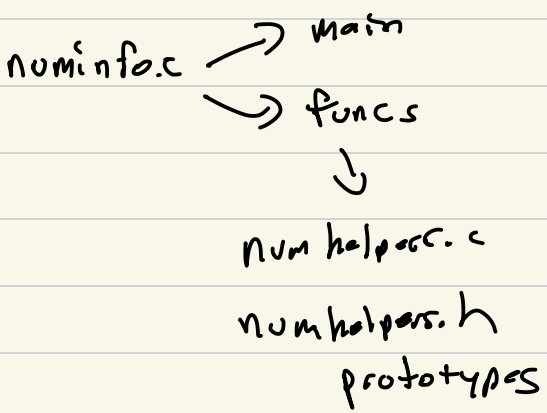
```
typedef struct config_st *config_p;

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

int foo(struct config_st *cp) {
```

Separate Compilation

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



numconv.c

```
#include <stdio.h>
    .
    .
    .
#include "numhelpers.c"
```

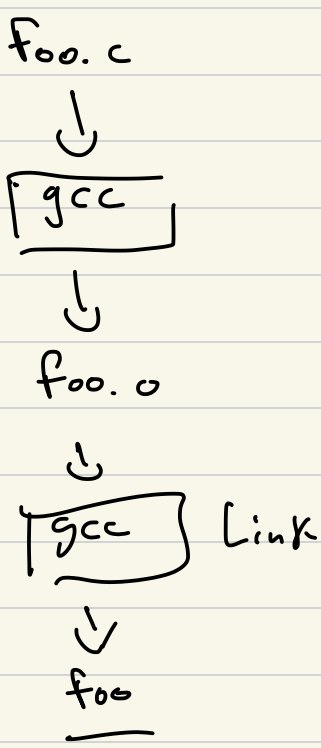
Makefile

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

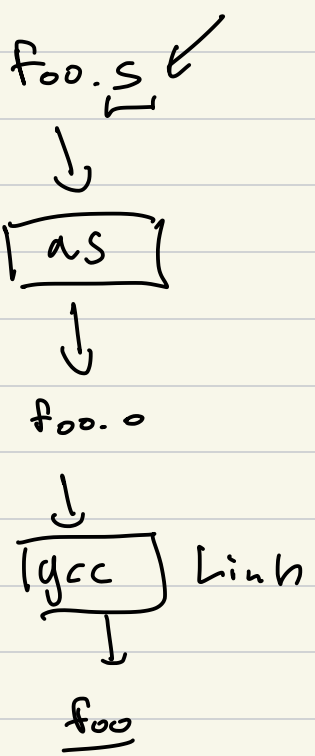
RISC-V Assembly Language

Assembly Language → human readable form of machine language machine code

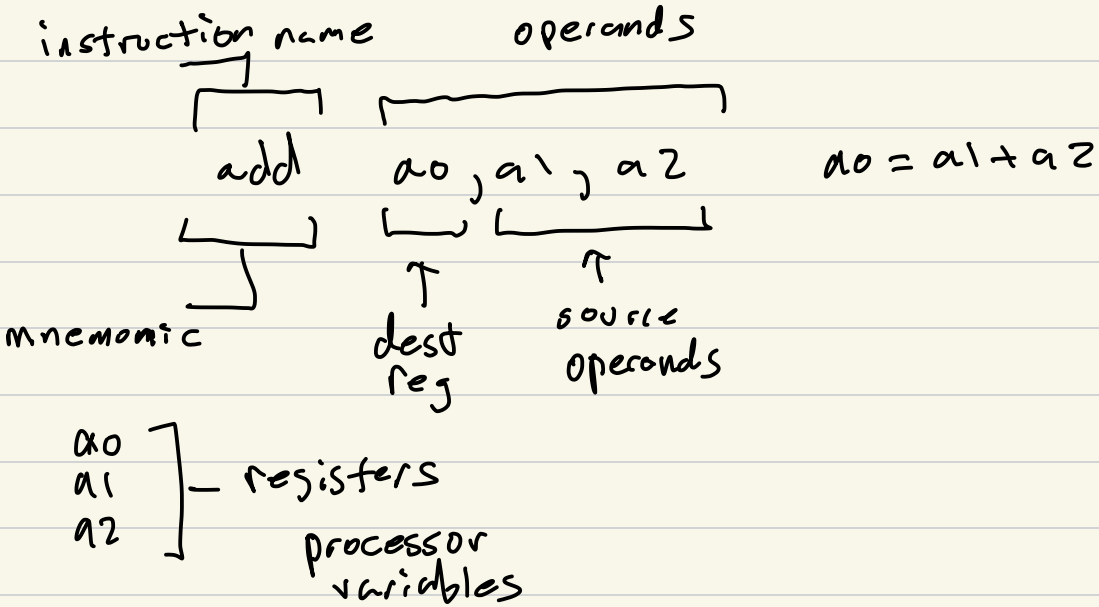
Compiling



Assembling



Vocabulary

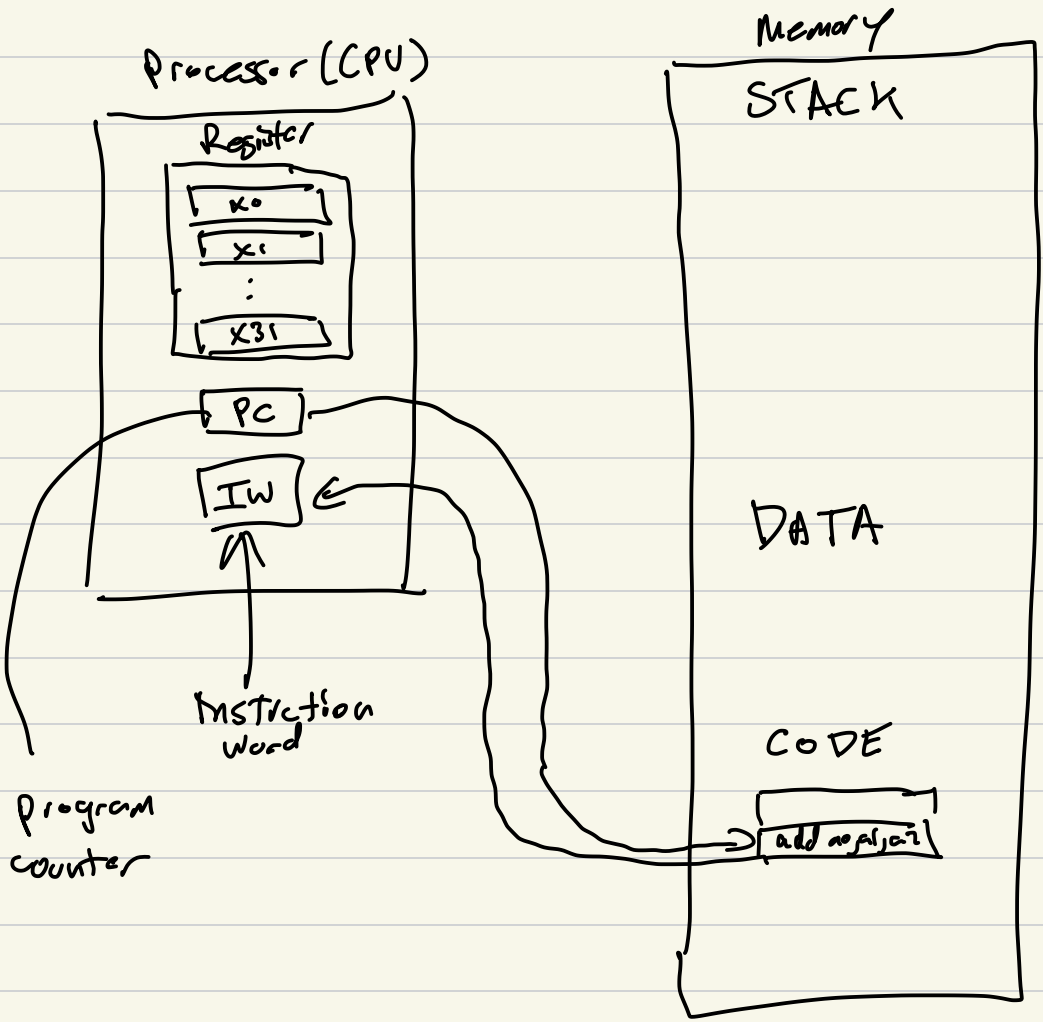


registers : 32
each register is 64 bits;

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

a0, a1, a2, ... arguments
t0, t1, t2, ... temp regs

Machine Code Execution Model



Assembly Source Components

labels
instructions
directives

RISC-V Assembly Types of Instructions

3 categories

1) Data Processing — add
2) Control — ret
3) Memory