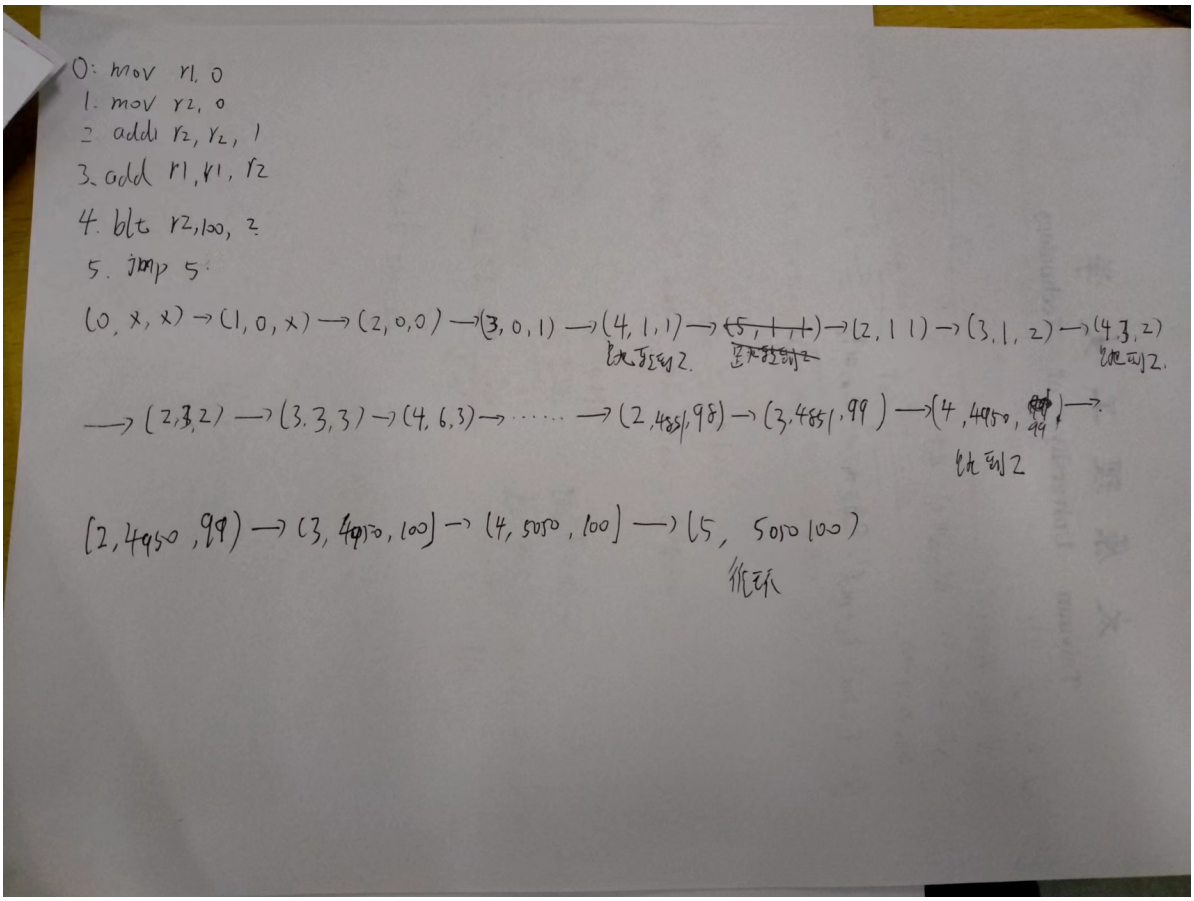
程序是个状态机



理解基础设施

次数：500

调试次数：$500 \times 0.9 = 450$

调试一次花费的时间：$20 \times 30s = 10mins$

使用GDB完成PA的时间：$450 \times 10mins = 4500mins$

使用sdb调试一次的时间：$20 \times 10s = 200s$

使用sdb完成PA的时间：$450 \times 200s = 1500mins$

综上：一个学期下来，节省了3000mins的时间，换而言之节省了50h。

RISC-V32（64）

指令有几种格式?

RV32I（基础指令集）中有RISBUJ六种。用于寄存器-寄存器操作的 R 类型指令，用 于短立即数和访存 load 操作的 I 型指令，用于访存 store 操作的 S 型指令，用于条件跳转操 作的 B 类型指令，用于长立即数的 U 型指令和用于无条件跳转的 J 型指令。

RV32C（压缩指令集扩展）中有CR、CI、CSS、CIW、CL、CS、CB、CJ八种（一半长度）

RV32M（乘法扩展）使用了RV32I的R指令格式

RV32F/D也是使用RV32I中的指令格式

RV32A（原子操作扩展）：R指令的funct7变为了funct5，用于增加了两位的标志位

综上：不考虑压缩指令集扩展，指令有六种格式；考虑压缩指令集扩展，指令有14种格式。需要注意的是，在RV32A扩展下，R型指令发生了一些变化。

LUI指令的行为是什么?

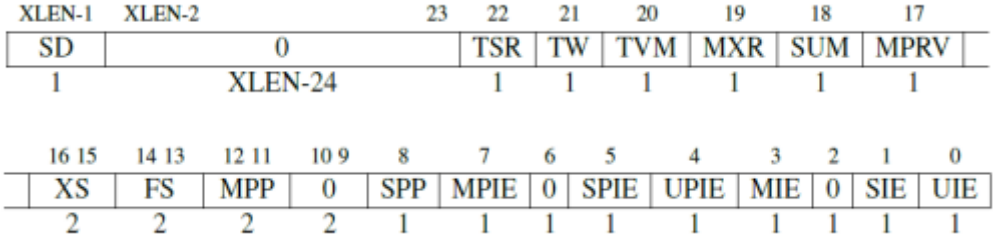
根据RISC-V Reference，lui是Load Upper Imm的缩写，即**高位立即数加载。**其C语言描述为：rd = imm << 12。U格式指令，定义于RV32I和RV64I。

行为：将符号位扩展的20位立即数（imm[31:12]）左移12位，并将低12位置零，写入x[rd]中（*rd*是 *5* 位的目的寄存器，指令运算的结果就存储 *rd* 中）

mstatus寄存器的结构是怎么样的?

mstatus是八个控制状态寄存器（CSR）的一种，是机器模式下异常处理的必要部分。

其结构如图所示【RISC-V READER page 101】：



XLEN-1	XLEN-2 … 23	22	21	20	19	18	17
SD	0	TSR	TW	TVM	MXR	SUM	MPRV
1	XLEN-24	1	1	1	1	1	1

16 15	14 13	12 11	10 9	8	7	6	5	4	3	2	1	0
XS	FS	MPP	0	SPP	MPIE	0	SPIE	UPIE	MIE	0	SIE	UIE
2	2	2	2	1	1	1	1	1	1	1	1	1

SHELL命令

```
marinatoo@marinatoo:~/ysyx-workbench/nemu$ make count
find . -name '*.c'|xargs wc -l|grep "total"|awk '{print }'
  44930 total
find . -name '*.h'|xargs wc -l|grep "total"|awk '{print }'
  34342 total
marinatoo@marinatoo:~/ysyx-workbench/nemu$ make count_without_blank
find . -name "*.c"|xargs cat|grep -v ^$ |wc -l
38284
find . -name "*.h"|xargs cat|grep -v ^$ |wc -l
31780
marinatoo@marinatoo:~/ysyx-workbench/nemu$
```

如图所示

RTFM

解释如下：（来自于PA文档）

- `-Wall`, `-Werror`: 在编译时刻把潜在的fault直接转变成failure. 这种工具的作用很有限, 只能寻找一些在编译时刻也觉得可疑的fault, 例如 `if (p = NULL)`. 不过随着编译器版本的增强, 编译器也能发现代码中的一些未定义行为open in new window. 这些都是免费的午餐, 不吃就真的白白浪费了.