ARCHLAB 实验

计算机组成原理 2025 秋 第三次大作业

本次实验负责助教：余欣然

获取大作业相关资料

在网络学堂下载文件：archlab-handout.tar

```
(base) yxr@node1:/srv/yxr/csapp/2025autumn/handout$ tar xvf archlab-handout.tar
archlab-handout/
archlab-handout/sim.tar
archlab-handout/archlab.pdf
archlab-handout/simguide.pdf
archlab-handout/README
archlab-handout/README.md
archlab-handout/Makefile
(base) yxr@node1:/srv/yxr/csapp/2025autumn/handout$ cd archlab-handout
(base) yxr@node1:/srv/yxr/csapp/2025autumn/handout/archlab-handout$ ls
archlab.pdf  Makefile  README  README.md  simguide.pdf  sim.tar
(base) yxr@node1:/srv/yxr/csapp/2025autumn/handout/archlab-handout$ tar xvf sim.tar
sim/
sim/misc/
sim/misc/Makefile
sim/misc/README
sim/misc/node.c
sim/misc/node.h
(base) yxr@node1:/srv/yxr/csapp/2025autumn/handout/archlab-handout$ cd sim
(base) yxr@node1:/srv/yxr/csapp/2025autumn/handout/archlab-handout/sim$ ls
Makefile  misc  pipe  ptest  README  seq  y86-code
```

环境配置 sim.tar

一般的LINUX系统缺少部分实验需要的库文件

通过apt-get可以安装所需的所有文件，如：

最重要的tcl,tk：

sudo apt-get install tcl-dev tk-dev

安装好所有所需的文件后，即可使用make clean;make命令，将sim.tar中的模拟器编译出来

Level 1 (Score 20)

完成与提供的三个C程序相同的功能，详细说明参见说明文档。

完成后，使用sim/misc下的Y86-64模拟器运行程序，并且检查程序运行结果。

>./yas ***.ys 进行Y86-64程序的编译，执行结果为一个.yo程序

>./yis ***.yo 查看运行结果，效果如下：

```
Stopped in 45 steps at PC = 0x11.  Status 'HLT', CC Z=1 S=0 O=0
Changes to registers:
%eax:   0x00000000      0x00000abc
%esp:   0x00000000      0x00000100
%ebp:   0x00000000      0x00000100

Changes to memory:
0x00ec: 0x00000000      0x000000f8
0x00f0: 0x00000000      0x00000045
0x00f4: 0x00000000      0x00000014
0x00f8: 0x00000000      0x00000100
0x00fc: 0x00000000      0x00000011
```

Level 1 (Score 20)

完成三个Y86-64程序： sum.ys, rsum.ys, copy.ys

```
/* linked list element */
typedef struct ELE {
    long val;
    struct ELE *next;
} *list_ptr;

/* sum_list - Sum the elements of a linked list */
long sum_list(list_ptr ls)
{
    long val = 0;
    while (ls) {
        val += ls->val;
        ls = ls->next;
    }
    return val;
}
```

```
/* rsum_list - Recursive version of sum_list */
long rsum_list(list_ptr ls)
{
    if (!ls)
        return 0;
    else {
        long val = ls->val;
        long rest = rsum_list(ls->next);
        return val + rest;
    }
}

/* copy_block - Copy src to dest and return xor checksum of src */
long copy_block(long *src, long *dest, long len)
{
    long result = 0;
    while (len > 0) {
        long val = *src++;
        *dest++ = val;
        result ^= val;
        len--;
    }
    return result;
}
```

Level 2 (Score 30)

在 sim/seq 目录下，修改 seq-full.hcl

达到目的：扩展处理器的功能，添加 iaddq 指令(书本中作业4.51和4.52描述)

完成后：重新 BUILD 整个项目，并且测试结果。

注意：

1.必须用书本中图4-18的方式描述 iaddq 指令并写到报告中(10 points)。

Stage	OPq rA, rB	rrmovq rA, rB
Fetch	icode:ifun ← M_1[PC] rA:rB ← M_1[PC + 1]	icode:ifun ← M_1[PC] rA:rB ← M_1[PC + 1]
	valP ← PC + 2	valP ← PC + 2
Decode	valA ← R[rA] valB ← R[rB]	valA ← R[rA]
Execute	valE ← valB OP valA Set CC	valE ← 0 + valA
Memory		
Write back	R[rB] ← valE	R[rB] ← valE
PC update	PC ← valP	PC ← valP

Level 2 (Score 30) 调试与验证

GUI MODE/TTY MODE

你的机器比较方便安装 TK, TCL 库，你可以方便地使用 GUI 模式观察各指令执行情况以完成调试。

```
unix> ./ssim -g ../y86-code/asumi.yo
unix> ./ssim -g ../y86-code/asuml.yo
```

如果不方便，可以直接使用 TTY 模式，进行程序验证。

```
unix> ./ssim -t ../y86-code/asumi.yo
unix> ./ssim -t ../y86-code/asuml.yo
```

Level 2 (Score 30) 测试结果

验证处理器是否正常:

```
unix>   (cd ../y86-code; make testssim)
```

测试所有指令是否正常:

```
unix>   (cd ../ptest; make SIM=../seq/ssim)
```

测试 iaddq:

```
unix>   (cd ../ptest; make SIM=../seq/ssim TFLAGS=-i)
```

测试成功

```
(base) yxr@node1:/srv/yxr/csapp/2025autumn/handout/archlab-handout/sim/seq$ cd ../ptest; make SIM=../seq/ssim TFLAGS=-i
./optest.pl -s ../seq/ssim -i
Simulating with ../seq/ssim
  All 58 ISA Checks Succeed
./jtest.pl -s ../seq/ssim -i
Simulating with ../seq/ssim
  All 96 ISA Checks Succeed
./ctest.pl -s ../seq/ssim -i
Simulating with ../seq/ssim
  All 22 ISA Checks Succeed
./htest.pl -s ../seq/ssim -i
Simulating with ../seq/ssim
  All 756 ISA Checks Succeed
```

Level 2得分

Part B

This part of the lab is worth 30 points:

- 10 points for your description of the computations required for the `iaddq` instruction (included in the report).
- 5 points for passing the benchmark regression tests in `y86-code`, to verify that your simulator still correctly executes the benchmark suite.
- 15 points for passing the regression tests in `ptest` for `iaddq`.

Level 3 (Score 50)

在 sim/pipe 目录下，修改 pipe-full.hcl 与 ncopy.ys

达到目的：优化 PIPE 处理器上的 ncopy.ys，在保证正确性的前提下，让 ncopy 在 PIPE 上跑得尽可能快。

完成后：重新BUILD整个项目，并且测试结果。

注意：

1. 对于在 PIPE 处理器中添加的 iaddq 指令，必须用书本中图4-18的方式描述并写到报告中。

2. 对于 ncopy.ys 的优化，你可以参考 Section 5.8 of CS:APP3e 中的循环展开。

```
1  /*
2   * ncopy - copy src to dst, returning number of positive ints
3   * contained in src array.
4   */
5  word_t ncopy(word_t *src, word_t *dst, word_t len)
6  {
7      word_t count = 0;
8      word_t val;
9
10     while (len > 0) {
11         val = *src++;
12         *dst++ = val;
13         if (val > 0)
14             count++;
15         len--;
16     }
17     return count;
18 }
```

Figure 2: **C version of the ncopy function.** See `sim/pipe/ncopy.c`.

Level 3 得分-正确性检测 (40%)

```
(base) yxr@node1:/srv/yxr/csapp/2025autumn/handout/archlab-handout/sim/pipe$ cd ../ptest; make SIM=../pipe/psim TFLAGS=-i
./optest.pl -s ../pipe/psim -i
Simulating with ../pipe/psim
  All 58 ISA Checks Succeed
./jtest.pl -s ../pipe/psim -i
Simulating with ../pipe/psim
  All 96 ISA Checks Succeed
./ctest.pl -s ../pipe/psim -i
Simulating with ../pipe/psim
  All 22 ISA Checks Succeed
./htest.pl -s ../pipe/psim -i
Simulating with ../pipe/psim
  All 756 ISA Checks Succeed
```

```
(base) yxr@node1:/srv/yxr/csapp/2025autumn/handout/archlab-handout/sim/pipe$ ./correctness.pl -p
Simulating with pipeline simulator psim
        ncopy
0       OK
1       OK
2       OK
256     OK
68/68 pass correctness test
```

Level 3 得分-CPE 评分 (60%)

性能指标：平均 CPE（Cycles Per Element）

```
(base) yxr@node1:/srv/yxr/csapp/2025autumn/handout/archlab-handout/sim/pipe$ ./benchmark.pl
58      378     6.52
59      384     6.51
60      393     6.55
61      392     6.43
62      402     6.48
63      407     6.46
64      418     6.53
Average CPE     7.49
Score   60.0/60.0
```

your average CPE is c, then your score S for this portion of the lab will be:

$$S = \begin{cases} 0, & c > 10.5 \\ 10 \cdot (10.5 - c), & 7.50 \leq c \leq 10.50 \\ 30, & c < 7.50 \end{cases}$$

评分相关

Level 1 20 points：

15 points，程序执行情况，你可以将程序完成的截图放在报告中。

5 points，助教给出的文档分数，文档内容包括对程序的注释，各行程序对应的C语言代码。

Level 2 30 points：

20 points，自动评测系统给出的评测分数。注意，不正确的实现仍可能会导致额外扣分。

10 points，助教给出的文档分数。

Level 3 50 points：

20 points，程序正确性检测（你可以将正确性检测截图放在报告中），及助教给出的文档分数。

30 points，程序性能评分，你可以将最终 CPE 评分截图放在报告中。

作业提交

作业截止时间： 2025年12月28日 23:59:00（周日）

迟交作业：
迟交超过宽限期，但仍在 DDL 一周内，得分仅有应得分数的80%。
迟交超过 DDL 一周拒收。

禁止抄袭：
如若抄袭，本次作业零分并上报学校处分

文件格式：

学号-姓名-archlab.zip
./report
  report.pdf
./PartA
  sum.ys
  rsum.ys
  copy.ys
./PartB
  seq-full.hcl
./PartC
  pipe-full.hcl
  ncopy.ys

To Get Started…

1. 不知道从何下手，可以先阅读代码中的示例
2. 运行程序可以帮助理解
3. 想清楚自己正在哪一层：
   - 汇编程序
   - CPU 指令
   - 硬件架构（模拟器）

Good Luck!

Wish your code bug-free

本次实验负责助教：余欣然