

计组第一次作业

计 71 张程远 2017011429

1.1

按照文档内容，先编译 Kernel，再打开 Term，即可测试 Term 中的命令。

1.2

直接使用文件中给出的示例程序 fib.s 即可。Term 中的代码不支持 label 功能，所以需要手动计算 offset 代替 label。由于每次使用 \$1 会报出 warning，因此我把所有的寄存器编号加了 1，代码和运行结果如下图：

```
C:\Windows\system32\cmd.exe
connecting to 127.0.0.1:6666...connected
MONITOR for MIPS32 - initialized.
>> A
>>addr: 0x80100000
one instruction per line, empty line to end.
[0x80100000] addiu $2, $0, 0x1
[0x80100004] addiu $3, $0, 0x1
[0x80100008] addiu $4, $0, 0
[0x8010000c] addiu $5, $0, 10
[0x80100010] addiu $6, $6, 0x8040
[0x80100014] sll $6, $6, 16
[0x80100018] sw $2, 0($6)
[0x8010001c] addiu $6, $6, 4
[0x80100020] sw $3, 0($6)
[0x80100024] addu $4, $3, $2
[0x80100028] addiu $6, $6, 4
[0x8010002c] sw $4, 0($6)
[0x80100030] addiu $2, $3, 0
[0x80100034] addiu $3, $4, 0
[0x80100038] addiu $5, 0xffff
[0x8010003c] bgtz $5, -28
[0x80100040] nop
[0x80100044] jr $31
[0x80100048] nop
[0x8010004c]
>> G
>>addr: 0x80100000
elapsed time: 0.000s
>> D
>>addr: 0x80400000
>>num: 40
0x80400000: 0x00000001
0x80400004: 0x00000001
0x80400008: 0x00000002
0x8040000c: 0x00000003
0x80400010: 0x00000005
0x80400014: 0x00000008
0x80400018: 0x0000000d
0x8040001c: 0x00000015
0x80400020: 0x00000022
0x80400024: 0x00000037
```

1.3

此程序需要使用 Syscall 输出字符到屏幕，因此需要使用带中断功能的 Kernel。整体逻辑就是判断字符输出是否结束，如果没有结束则令 ascii 码加 1 并继续输出。首先按照文档编译 Kernel，然后测试代码，结果如下：

```

C:\Windows\system32\cmd.exe
connecting to 127.0.0.1:6666...connected
MONITOR for MIPS32 - initialized.
>> A
>>addr: 0x80100000
one instruction per line, empty line to end.
[0x80100000] addiu $2, $0, 30
[0x80100004] addiu $4, $0, 0x20
[0x80100008] addiu $5, $0, 0x7f
[0x8010000c] addiu $4, $4, 0x1
[0x80100010] syscall
[0x80100014] bne $4, $5, -12
[0x80100018] nop
[0x8010001c] jr $31
[0x80100020] nop
[0x80100024]
>> G
>>addr: 0x80100000
!#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNQRSTUvwxyz[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
elapsed time: 0.002s
>>

```

1.4

提示用两个寄存器存储 64 位整数，因此运算过程中的每个整数都需要两个寄存器。代码中 t1 使用\$2、\$3 存储，t2 使用\$4、\$5 存储，t3=t1+t2 使用\$6、\$7 存储。偶数寄存器存储低 32 位，奇数寄存器存储高 32 位。当产生进位时，\$6 的值应该比\$2 和\$4 的值要小，因此使用 sltu 指令判断\$6 和\$2 的大小关系，产生的标记值存入\$9 中，如果判定产生进位则让\$7 的值加 1。代码运行结果如下：

```

C:\Windows\system32\cmd.exe
connecting to 127.0.0.1:6666...connected
MONITOR for MIPS32 - initialized.
>> A
>>addr: 0x80100000
one instruction per line, empty line to end.
[0x80100000] addiu $2, $0, 0x1
[0x80100004] addiu $4, $0, 0x1
[0x80100008] addiu $8, $0, 58
[0x8010000c] addiu $10, $10, 0x8040
[0x80100010] sll $10, $10, 16
[0x80100014] addu $6, $4, $2
[0x80100018] addu $7, $5, $3
[0x8010001c] sltu $9, $2, $6
[0x80100020] bgtz $9, 8
[0x80100024] nop
[0x80100028] addiu $7, $7, 0x1
[0x8010002c] addiu $2, $4, 0
[0x80100030] addiu $3, $5, 0
[0x80100034] addiu $4, $6, 0
[0x80100038] addiu $5, $7, 0
[0x8010003c] addiu $8, 0xffff
[0x80100040] bgtz $8, -48
[0x80100044] nop
[0x80100048] sw $6, 0($10)
[0x8010004c] sw $7, 4($10)
[0x80100050] jr $31
[0x80100054] nop
[0x80100058]
>> G
>>addr: 0x80100000
elapsed time: 0.000s
>> D
>>addr: 0x80400000
>>num: 8
0x80400000: 0x6c8312d0
0x80400004: 0x00000168
>>

```

其中斐波那契数列的第 60 项为 1548008755920, 换算为 16 进制结果为 1686c8312d0。