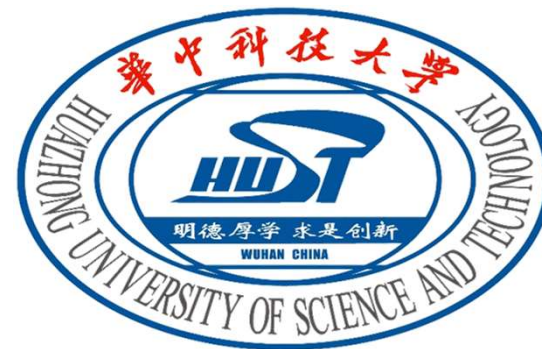


# 微机原理与接口技术

## 计算机工作原理

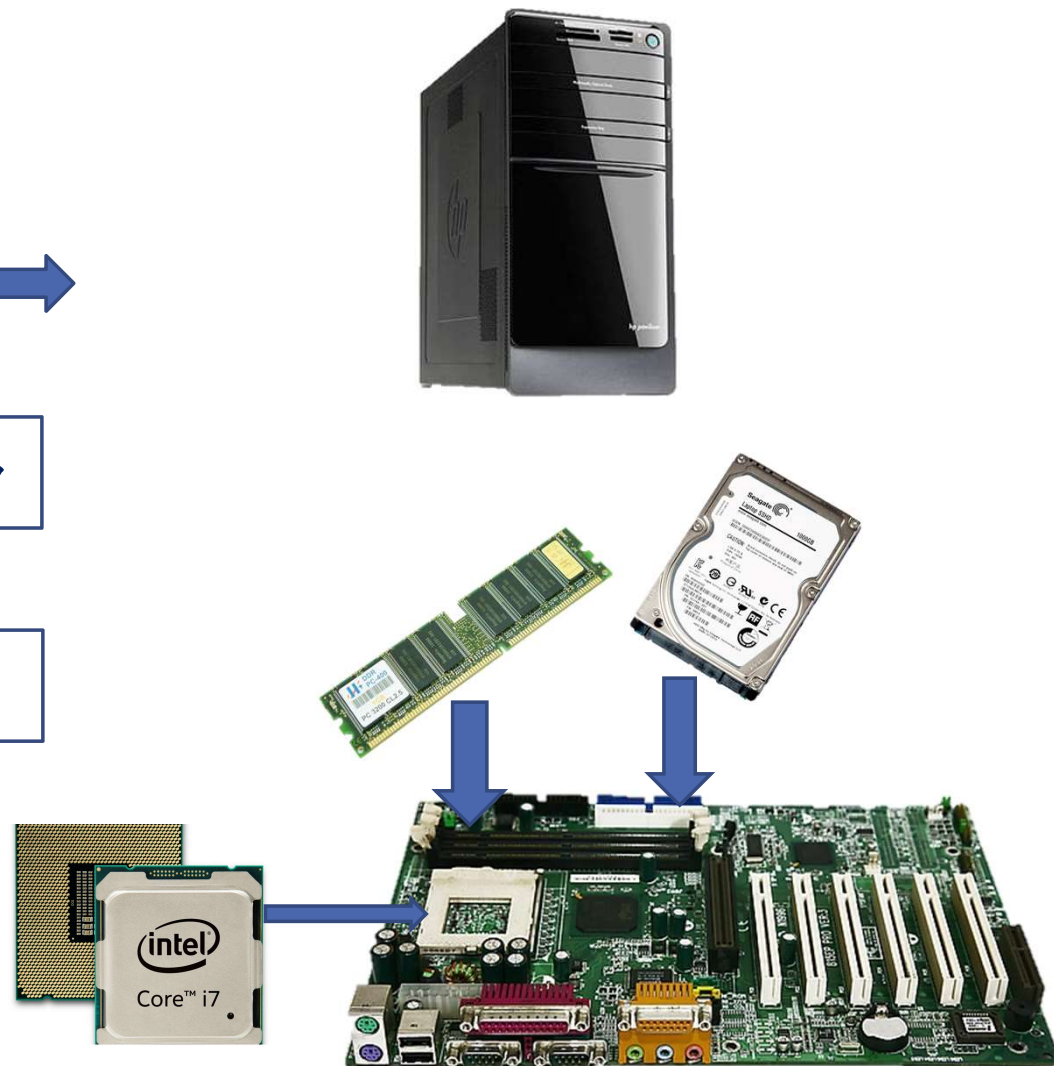
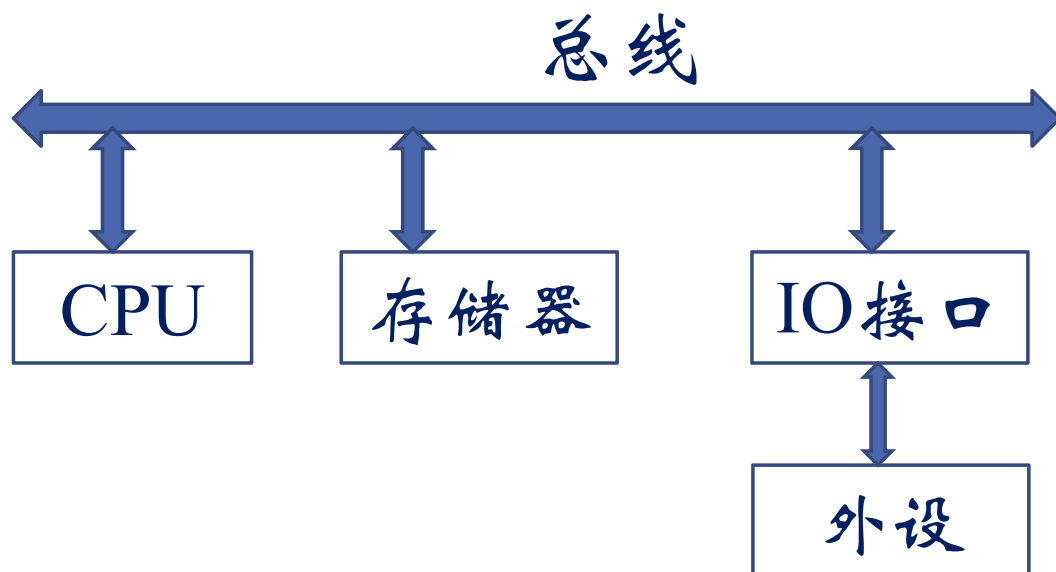
---

华中科技大学 左冬红



华中科技大学电子信息与通信学院 左冬红

# 计算机结构




# 术语

- 程序
  - 实现既定任务的指令序列
  - C语言程序、python语言程序……
- 指令
  - 计算机能识别并能执行的基本操作命令
  - 机器指令、汇编指令
- 指令集
  - 计算机所能执行的全部指令
  - MIPS32指令集、intel指令集……

# 计算机解决问题的过程

## • 编写程序

- 将任务分解成计算机能识别并能执行的基本操作命令，这些基本操作命令按一定顺序排列起来预先存储到存储器中



```
swap.c - 记事本
文件(F) 编辑(E) 格式(O) 查看(V)
帮助(H)
int main()
{
    int word[2]={80,100};
    int tmp;
    tmp=word[0];
    word[0]=word[1];
    word[1]=tmp;
    return 0;
}
```

语句

mips-img-elf-gcc.exe -S swap.c

mips-img-elf-gcc.exe swap.c

a.out

mips-img-elf-objdump.exe a.out

```
swap_mars.s - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

.text
.globl    main
main:
    addiu    $sp,$sp,-24
    sw       $fp,20($sp)
    move     $fp,$sp
    li       $2,$2,80
    sw       $2,4($fp)
    li       $2,$2,100
    sw       $2,8($fp)
    lw       $2,4($fp)
    sw       $2,0($fp)
    lw       $2,8($fp)
    sw       $2,4($fp)
    lw       $2,0($fp)
    sw       $2,8($fp)
    move     $2,$0
    move     $sp,$fp
    lw       $fp,20($sp)
    addiu    $sp,$sp,24
    jr       $31
```

汇编指令

# 计算机解决问题过程

a.out

mips-img-elf-objdump.exe a.out

机器指令

```
swapobj.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
Contents of section .text:
4000d4 3c040041 3c020041 24840318 2442031b <..A<..A$..$B..
4000e4 00441023 24030006 18620005 3c190000 .D.#$...b.<...
4000f4 27390000 db200002 00000000 d8190000 '9... ..
400104 d81f0000 3c040041 3c050041 24840318 ....<..A<..A$...
400114 24a50318 00a42823 00052883 000517c2 $.....(#{..(....
400124 00452821 00052843 d8a00005 3c190000 .E(!..(C....<...
400134 27390000 db200002 00000000 d8190000 '9... ..
400144 d81f0000 27bdf8d8 afb1001c afb00020 ....'.....
400154 3c110041 9222031c afb30014 afb20018 <..A.".....
400164 14400017 afbf0024 3c100041 3c130041 .@.....$<..A<..A
400174 2602030c 3c100041 26100310 02028023 &...<..A&.....#
400184 3c020041 00108083 2452030c 2610ffff <..A....$R..&...
400194 8e620320 1c500011 00000000 ebffffcc .b..P.....
4001a4 3c020000 24420000 d8400003 3c040040 <...$B...@...<...@
4001b4 24840300 ebffff91 24020001 a222031c $......$...."..
4001c4 8fb30014 8fb20018 8fb1001c 8fb00020 .....
4001d4 8fbf0024 03e00009 27bd0028 24420001 ...$....'..($B..
4001e4 ae620320 00521045 8c420000 f8020000 .b..R.E.B.....
4001f4 1000ffe8 8e620320 27bdffe8 3c020000 ....b.'...<...
400204 24420000 10400006 afbf0014 3c050041 $B...@.....<..A
400214 3c040040 24a50324 24840300 ebffff77 <...@$..$$.w
400224 3c020041 24440314 8c820000 d8400005 <..A$D.....@..
400234 3c020000 24420000 d8400002 00000000 <...$B...@.....
400244 f8020000 8fbf0014 27bd0018 cbffffad .....
400254 27bdffe8 afbe0014 03a0f025 24020050 '.....%$..P
```



# MIPS模拟器运行程序

swap\_mars.s - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
.text
.globl main

main:
    addiu $sp,$sp,-24
    sw $fp,20($sp)
    move $fp,$sp
    li $2,80
    sw $2,4($fp)
    li $2,100
    sw $2,8($fp)
    lw $2,4($fp)
    sw $2,0($fp)
    lw $2,8($fp)
    sw $2,4($fp)
    lw $2,0($fp)
    sw $2,8($fp)
    move $2,$0
    move $sp,$fp
    lw $fp,20($sp)
    addiu $sp,$sp,24
    jr $31
```

Text Segment						
Bkpt	Address	Code	Basic			Source
<input type="checkbox"/>	0x00400000	0x27bdfef8	addiu \$29,\$29,0xffffffff	4:	addiu	\$sp,\$sp,-24
<input type="checkbox"/>	0x00400004	0xafbe0014	sw \$30,0x00000014(\$29)	5:	sw	\$fp,20(\$sp)
<input type="checkbox"/>	0x00400008	0x001df021	addu \$30,\$0,\$29	6:	move	\$fp,\$sp
<input type="checkbox"/>	0x0040000c	0x24020050	addiu \$2,\$0,0x00000050	7:	li	\$2,80
<input type="checkbox"/>	0x00400010	0xafc20004	sw \$2,0x00000004(\$30)	8:	sw	\$2,4(\$fp)
<input type="checkbox"/>	0x00400014	0x24020064	addiu \$2,\$0,0x00000064	9:	li	\$2,100
<input type="checkbox"/>	0x00400018	0xafc20008	sw \$2,0x00000008(\$30)	10:	sw	\$2,8(\$fp)
<input type="checkbox"/>	0x0040001c	0x8fc20004	lw \$2,0x00000004(\$30)	11:	lw	\$2,4(\$fp)
<input type="checkbox"/>	0x00400020	0xafc20000	sw \$2,0x00000000(\$30)	12:	sw	\$2,0(\$fp)
<input type="checkbox"/>	0x00400024	0x8fc20008	lw \$2,0x00000008(\$30)	13:	lw	\$2,8(\$fp)
<input type="checkbox"/>	0x00400028	0xafc20004	sw \$2,0x00000004(\$30)	14:	sw	\$2,4(\$fp)
<input type="checkbox"/>	0x0040002c	0x8fc20000	lw \$2,0x00000000(\$30)	15:	lw	\$2,0(\$fp)
<input type="checkbox"/>	0x00400030	0xafc20008	sw \$2,0x00000008(\$30)	16:	sw	\$2,8(\$fp)
<input type="checkbox"/>	0x00400034	0x00001021	addu \$2,\$0,\$0	17:	move	\$2,\$0
<input type="checkbox"/>	0x00400038	0x001ee821	addu \$29,\$0,\$30	18:	move	\$sp,\$fp
<input type="checkbox"/>	0x0040003c	0x8fb00014	lw \$30,0x00000014(\$29)	19:	lw	\$fp,20(\$sp)
<input type="checkbox"/>	0x00400040	0x27bd0018	addiu \$29,\$29,0x00000018	20:	addiu	\$sp,\$sp,24
<input type="checkbox"/>	0x00400044	0x03e00008	jr \$31	21:	jr	\$31

内存地址

机器指令



# MIPS模拟器运行程序

D:\01教学\微机原理\实验软件\Codewave.GNU.Tools.Package.2018.09-03.for.MIPS.IMG.Bare.Metal.Windows.x86\_64\bin\swap\_mars.s - MAR...

File Edit Run Settings Tools Help

Run speed 2 inst/sec

Edit Execute

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x27bdf8e8	addiu \$29,\$29,0xffffffff	4: addiu \$sp,\$sp,-24
	0x00400004	0xafbe0014	sw \$30,0x00000014(\$29)	5: sw \$fp,20(\$sp)
	0x00400008	0x001df021	addu \$30,\$0,\$29	6: move \$fp,\$sp
	0x0040000c	0x24020050	addiu \$2,\$0,0x00000050	7: li \$2,80
	0x00400010	0xafc20004	sw \$2,0x00000004(\$30)	8: sw \$2,4(\$fp)
	0x00400014	0x24020064	addiu \$2,\$0,0x00000064	9: li \$2,100
	0x00400018	0xafc20008	sw \$2,0x00000008(\$30)	10: sw \$2,8(\$fp)
	0x0040001c	0x8fc20004	lw \$2,0x00000004(\$30)	11: lw \$2,4(\$fp)
	0x00400020	0xafc20000	sw \$2,0x00000000(\$30)	12: sw \$2,0(\$fp)
	0x00400024	0x8fc20008	lw \$2,0x00000008(\$30)	13: lw \$2,8(\$fp)
	0x00400028	0xafc20004	sw \$2,0x00000004(\$30)	14: sw \$2,4(\$fp)
	0x0040002c	0x8fc20000	lw \$2,0x00000000(\$30)	15: lw \$2,0(\$fp)
	0x00400030	0xafc20008	sw \$2,0x00000008(\$30)	16: sw \$2,8(\$fp)
	0x00400034	0x00001021	addu \$2,\$0,\$0	17: move \$2,\$0
	0x00400038	0x001ee821	addu \$29,\$0,\$30	18: move \$sp,\$fp
	0x0040003c	0x8f8e0014	lw \$30,0x00000014(\$29)	19: lw \$fp,20(\$sp)
	0x00400040	0x27bd0018	addiu \$29,\$29,0x00000018	20: addiu \$sp,\$sp,24
	0x00400044	0x03e00008	jir \$31	21: jr \$31

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000

Coproc 1 Coproc 0

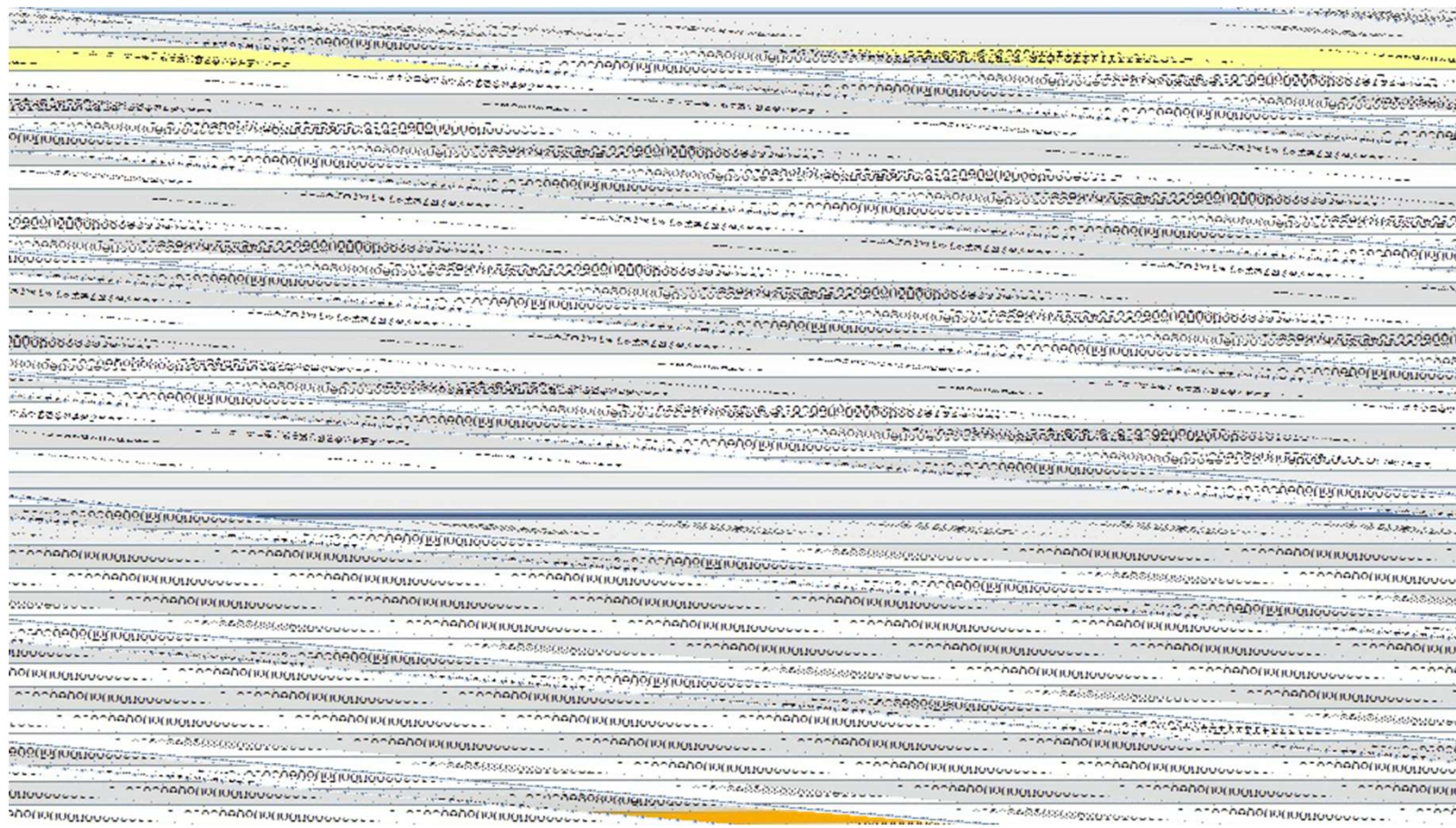
Registers

Name	Num...	Value
\$zero	0	0x00000000
\$at	1	0x00000000
\$v0	2	0x00000000
\$v1	3	0x00000000
\$a0	4	0x00000000
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x00000000
\$s1	17	0x00000000
\$s2	18	0x00000000
\$s3	19	0x00000000
\$s4	20	0x00000000
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7fffffc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x00400000
hi		0x00000000
lo		0x00000000



# MIPS模拟器运行程序

大家发现了什么？





# 计算机工作原理

- CPU中有专门的寄存器PC指示下一条即将运行指令的存储地址
- 指令是一条接着一条往下运行的

计算机如何知道执行什么功能呢？  
具体如何运行？

# 计算机工作原理

```
sw $2,8($fp)
lw $2,4($fp)
sw $2,0($fp)
lw $2,8($fp)
sw $2,4($fp)
lw $2,0($fp)
sw $2,8($fp)
```

汇编指令

101011	11110	00010	00000000000000001000
100011	11110	00010	0000000000000000100
101011	11110	00010	00000000000000000000
100011	11110	00010	00000000000000001000
101011	11110	00010	0000000000000000100
100011	11110	00010	00000000000000000000
101011	11110	00010	00000000000000001000

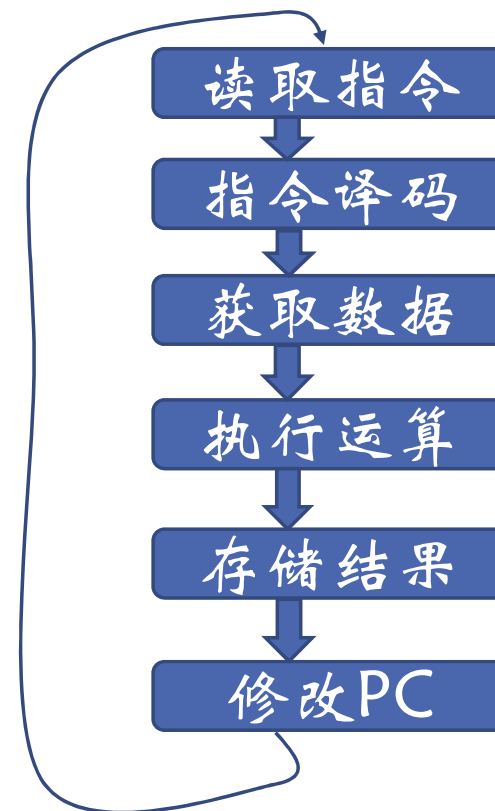
机器指令

# 计算机工作原理

sw\$2,8(\$fp)

101011 11110 00010 00000000000001000

lw \$2,4(\$fp)





## 小结

- 计算机程序运行前需存储起来——程序存储
- 然后再执行——程序执行
  - 读取指令
  - 译码指令
  - 获取操作数
  - 执行运算
  - 存储结果
  - 准备下一条指令（修改PC）