微机原理与接口技术

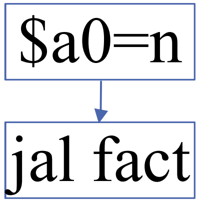
子程序递归调用

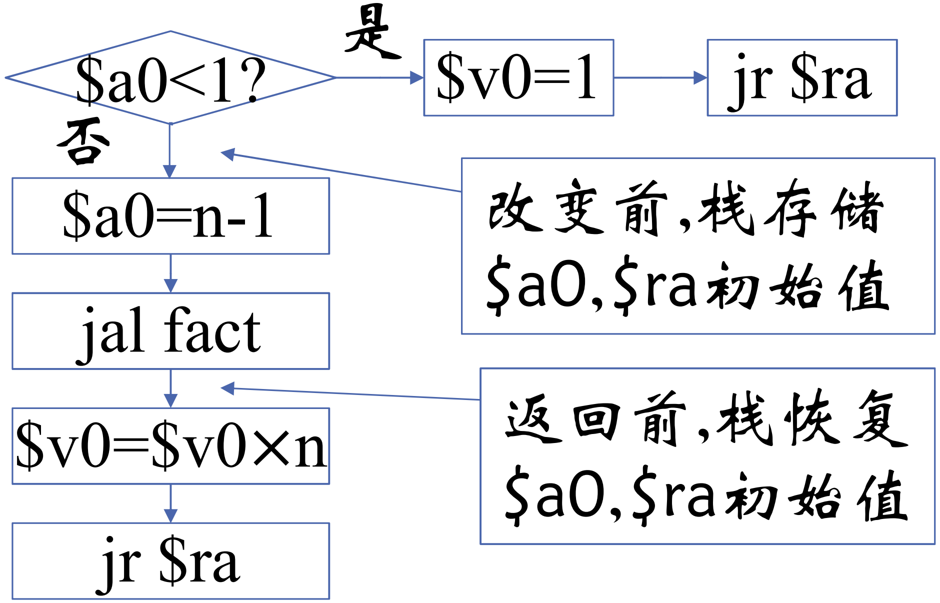
华中科技大学　左冬红

子程序递归调用

```
int fact（int n）
{
     if(n<1) return (1);
          else return (n*fact(n-1));
}
```

fact第一次被调用时



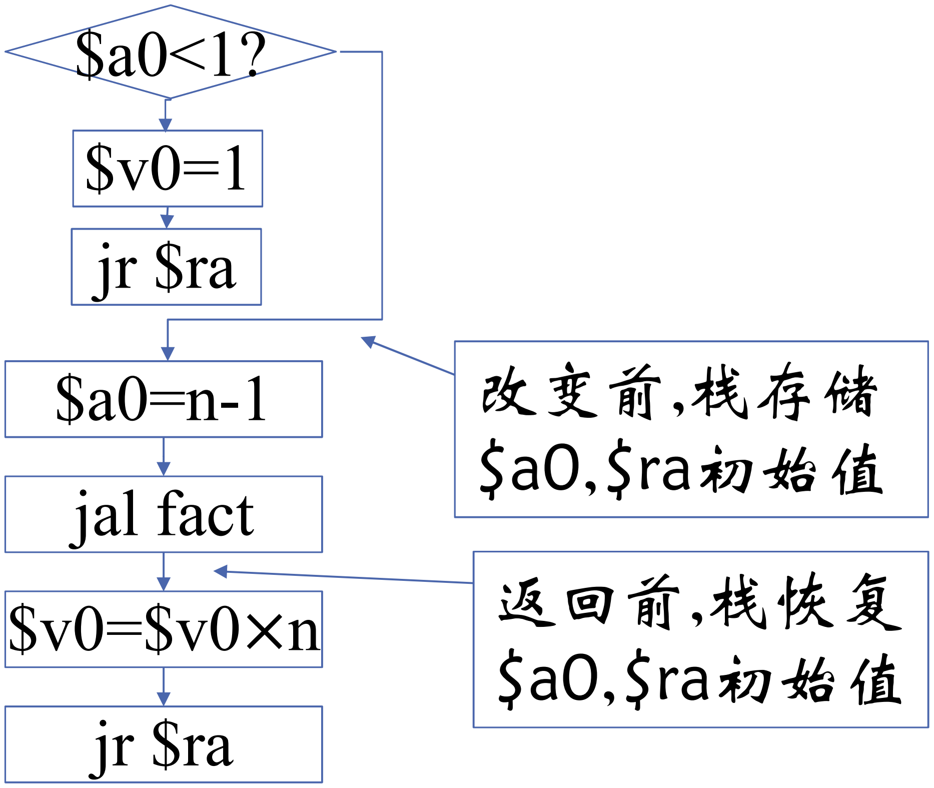


改变前,栈存储
$a0,$ra初始值

返回前,栈恢复
$a0,$ra初始值

$a0,$ra在子程序执行过程中发生改变

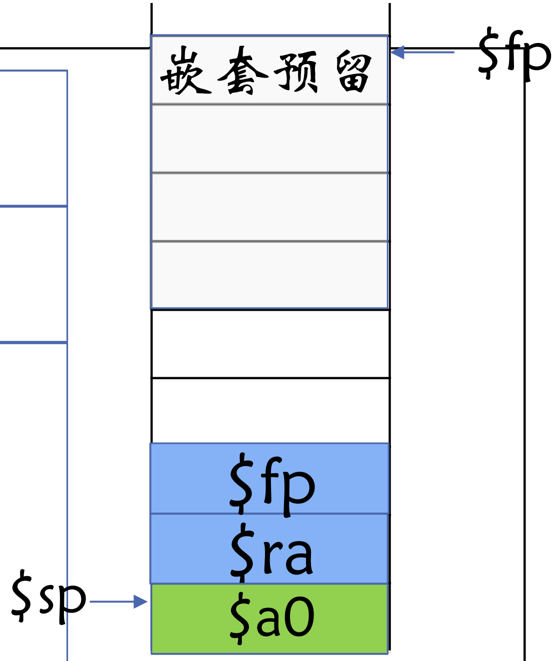
子程序嵌套调用



改变前，栈存储$a0,$ra初始值

返回前，栈恢复$a0,$ra初始值

```
fact: slti $t0,$a0,1
      beq $t0,$zero,L1
      addi $v0,$zero,1
      jr $ra
L1:   addi $sp,$sp,-32
      sw $ra,28($sp)
      sw $fp,24($sp)
      add $fp,$sp,0
      sw $a0,32($fp)
      addi $a0,$a0,-1
      jal fact
```



```
lw $a0,32($fp)
add $sp,$fp,0
lw $fp,24($sp)
lw $ra,28($sp)
addi $sp,$sp,32
mult $v0,$a0
mflo $v0
jr $ra
```

子程序嵌套调用执行时栈变化过程示例

```
int fact(int n)
{
    if(n<1) return (1);
        else return (n*fact(n-1));
}
int main()
{
    int c;
    c=fact(3);
    return 0;  &return
 }
```

fact(3) $ra=&return $a0=3

```
fact: slti $t0,$a0,1
      beq $t0,$zero,L1
      addi $v0,$zero,1
      jr $ra
  L1: addi $sp,$sp,-32
      sw $ra,28($sp)
      sw $fp,24($sp)
      add $fp,$sp,0
      sw $a0,32($fp)
      addi $a0,$a0,-1
      jal fact
```

&lw

```
lw $a0,32($fp)
add $sp,$fp,0
lw $fp,24($sp)
lw $ra,28($sp)
addi $sp,$sp,32
mult $v0,$a0
mflo $v0
jr $ra
```

fact(3)进入时,栈变化

```
fact: slti $t0,$a0,1
      beq $t0,$zero,L1
      addi $v0,$zero,1
      jr $ra
  L1: addi $sp,$sp,-32
      sw $ra,28($sp)
      sw $fp,24($sp)
      add $fp,$sp,0
      sw $a0,32($fp)
      addi $a0,$a0,-1
      jal fact
&lw   lw $a0,32($fp)
      add $sp,$fp,0
      lw $fp,24($sp)
      lw $ra,28($sp)
      addi $sp,$sp,32
      mult $v0,$a0
      mflo $v0
      jr $ra
```

进入fact(3)

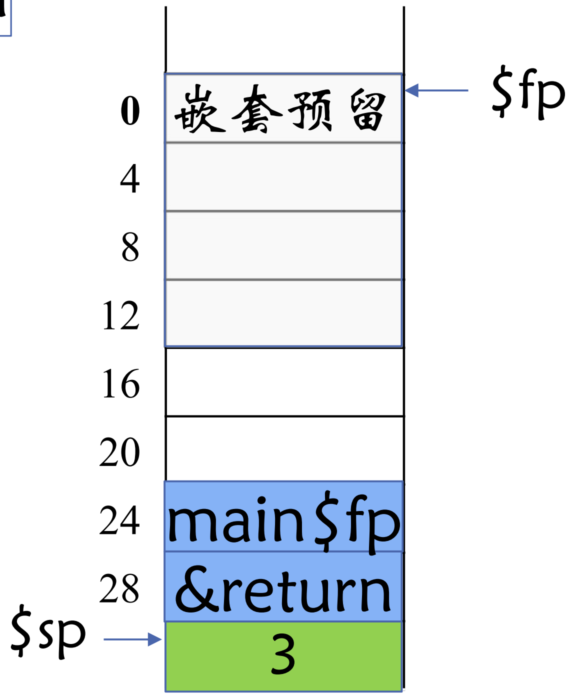
$ra=&return

$a0=3

进入fact(2)

$ra=&lw

$a0=2



0	嵌套预留	← $fp
4		
8		
12		
16		
20		
24	main$fp	
28	&return	
$sp →	3	

fact(2)进入时,栈变化

```
fact: slti $t0,$a0,1
      beq $t0,$zero,L1
      addi $v0,$zero,1
      jr $ra
 L1:  addi $sp,$sp,-32
      sw $ra,28($sp)
      sw $fp,24($sp)  &lw  lw $a0,32($fp)
      add $fp,$sp,0        add $sp,$fp,0
      sw $a0,32($fp)       lw $fp,24($sp)
      addi $a0,$a0,-1      lw $ra,28($sp)
      jal fact             addi $sp,$sp,32
                           mult $v0,$a0
                           mflo $v0
                           jr $ra
```

进入fact(2)

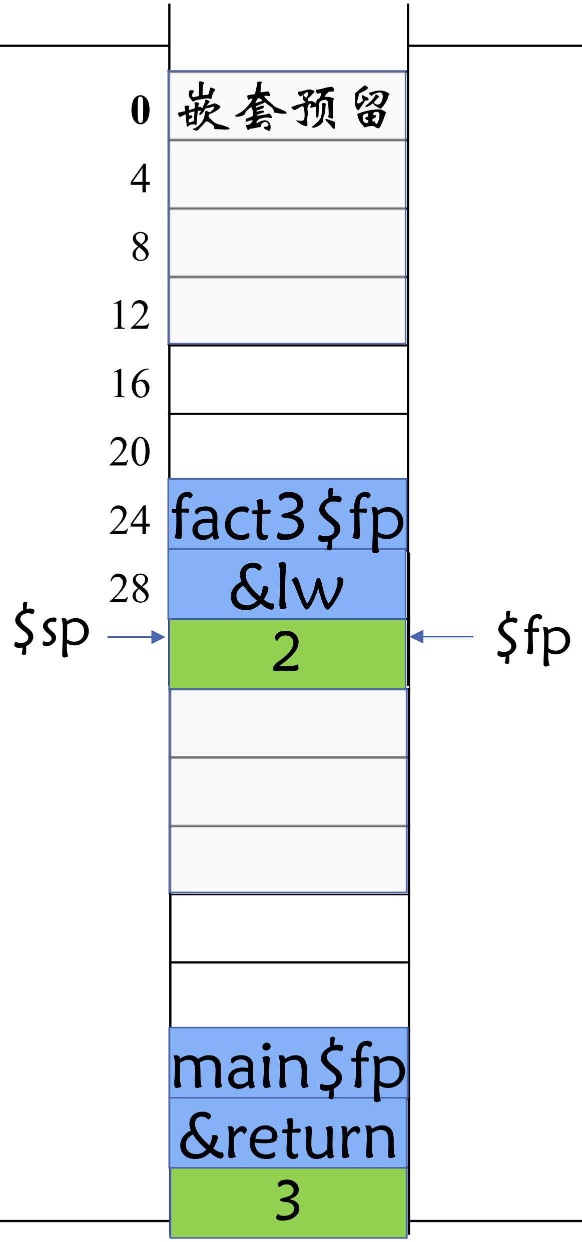
$ra=&lw

$a0=2

进入fact(1)

$ra=&lw

$a0=1



fact(1)进入时,栈变化

```
fact: slti $t0,$a0,1
      beq $t0,$zero,L1
      addi $v0,$zero,1
      jr $ra
  L1: addi $sp,$sp,-32
      sw $ra,28($sp)
      sw $fp,24($sp)
      add $fp,$sp,0
      sw $a0,32($fp)
      addi $a0,$a0,-1
      jal fact
```

&lw

```
lw $a0,32($fp)
add $sp,$fp,0
lw $fp,24($sp)
lw $ra,28($sp)
addi $sp,$sp,32
mult $v0,$a0
mflo $v0
jr $ra
```

进入fact(1)

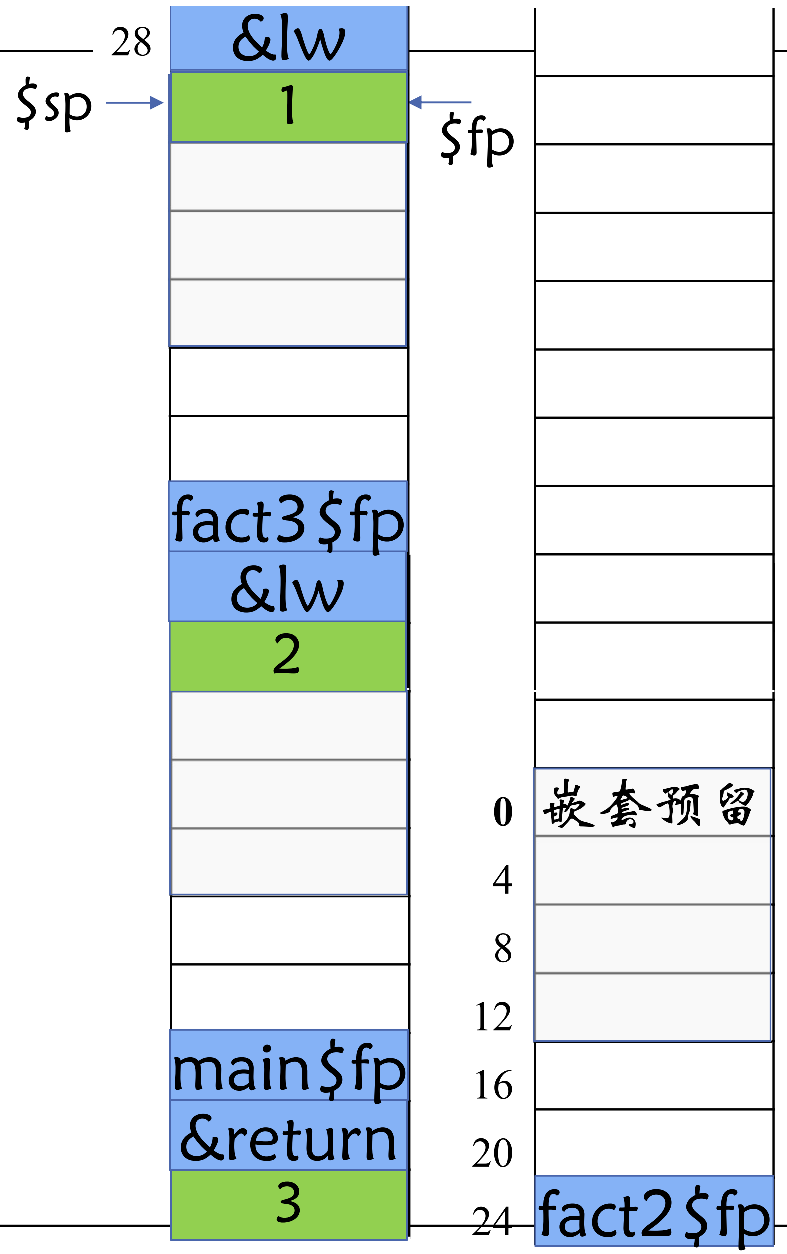
$ra=&lw

$a0=1

进入fact(0)

$ra=&lw

$a0=0



fact(0)进入

```
fact: slti $t0,$a0,1
      beq $t0,$zero,L1
      addi $v0,$zero,1
      jr $ra
  L1: addi $sp,$sp,-32
      sw $ra,28($sp)
      sw $fp,24($sp)
      add $fp,$sp,0
      sw $a0,32($fp)
      addi $a0,$a0,-1
      jal fact
```

进入fact(0)

$ra=&lw

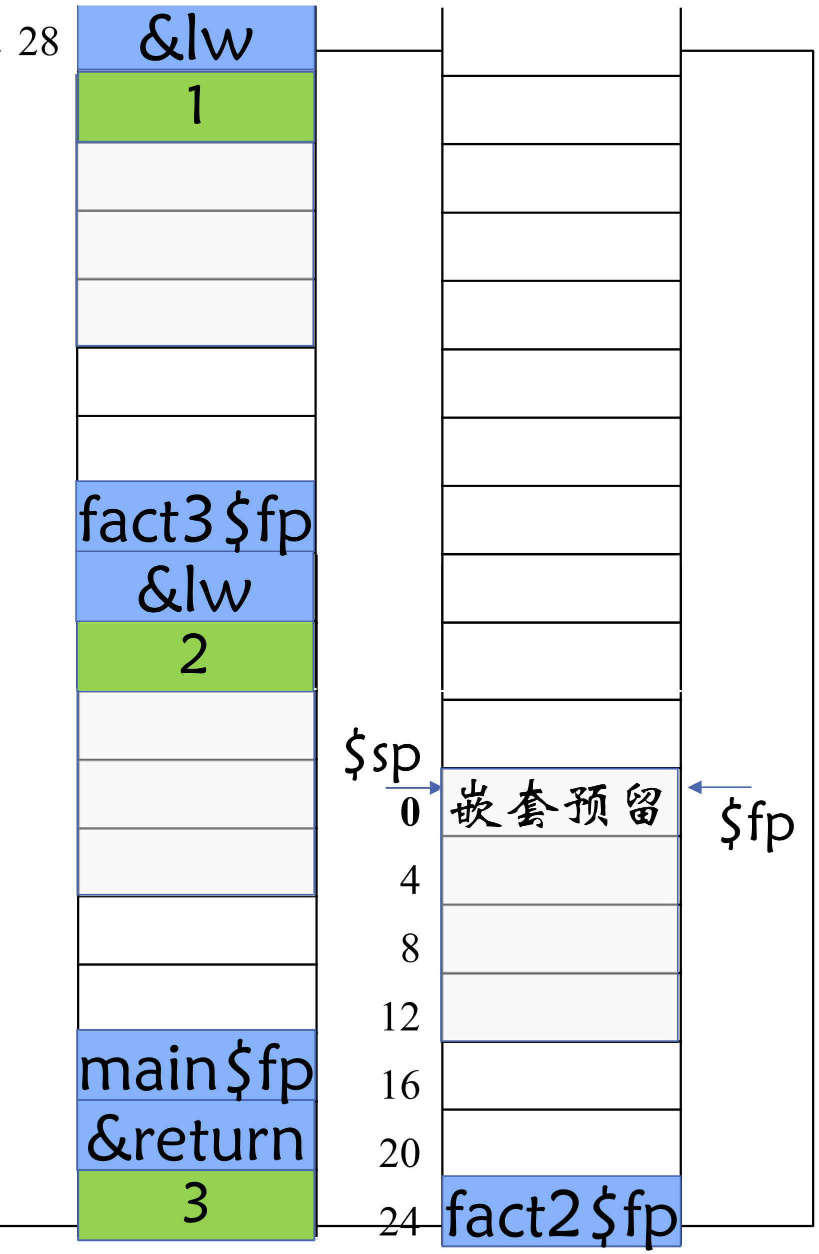
$a0=0

$v0=1

&lw

```
lw $a0,32($fp)
add $sp,$fp,0
lw $fp,24($sp)
lw $ra,28($sp)
addi $sp,$sp,32
mult $v0,$a0
mflo $v0
jr $ra
```

返回到fact(1)



fact(1)返回

```
fact: slti $t0,$a0,1
      beq $t0,$zero,L1
      addi $v0,$zero,1
      jr $ra
  L1: addi $sp,$sp,-32
      sw $ra,28($sp)
      sw $fp,24($sp)
      add $fp,$sp,0
      sw $a0,32($fp)
      addi $a0,$a0,-1
      jal fact
```

&lw

```
lw $a0,32($fp)
add $sp,$fp,0
lw $fp,24($sp)
lw $ra,28($sp)
addi $sp,$sp,32
mult $v0,$a0
mflo $v0
jr $ra
```

fact(1)返回

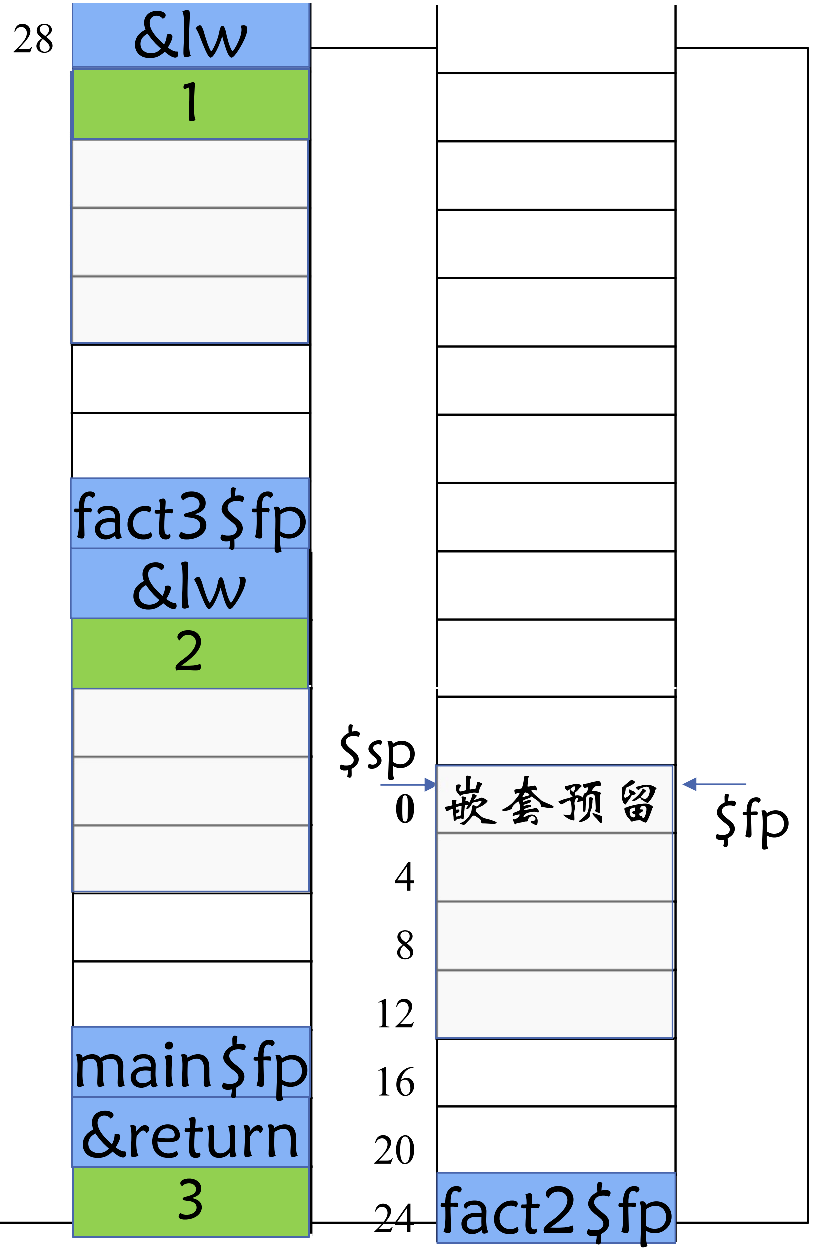
$v0=1

$a0=1

$ra=&lw

$v0=1

返回到fact(2)



fact(2)返回

```
fact: slti $t0,$a0,1
      beq $t0,$zero,L1
      addi $v0,$zero,1
      jr $ra
  L1: addi $sp,$sp,-32
      sw $ra,28($sp)
      sw $fp,24($sp)
      add $fp,$sp,0
      sw $a0,32($fp)
      addi $a0,$a0,-1
      jal fact
```

&lw

```
lw $a0,32($fp)
add $sp,$fp,0
lw $fp,24($sp)
lw $ra,28($sp)
addi $sp,$sp,32
mult $v0,$a0
mflo $v0
jr $ra
```

fact(2)返回

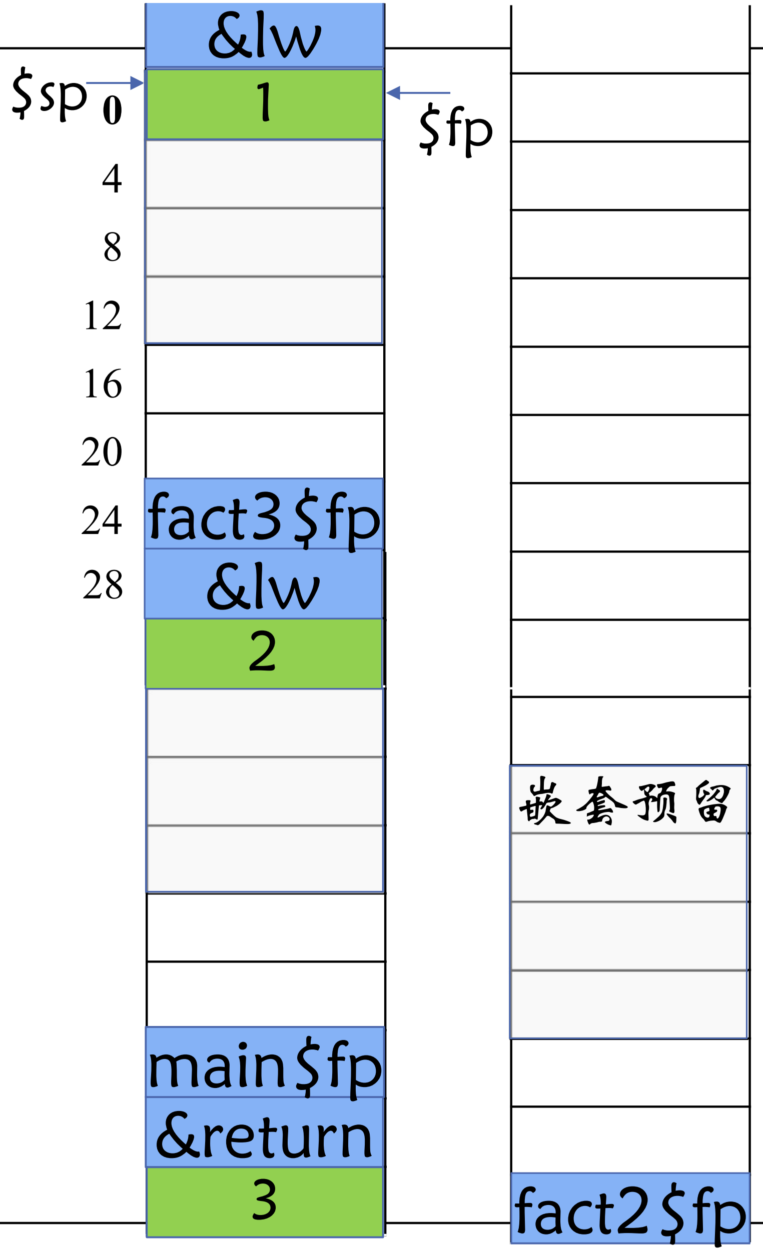
$v0=1

$a0=2

$ra=&lw

$v0=2

返回到fact(3)



fact(3)返回

```
fact: slti $t0,$a0,1
      beq $t0,$zero,L1
      addi $v0,$zero,1
      jr $ra
  L1: addi $sp,$sp,-32
      sw $ra,28($sp)
      sw $fp,24($sp)
      add $fp,$sp,0
      sw $a0,32($fp)
      addi $a0,$a0,-1
      jal fact
```

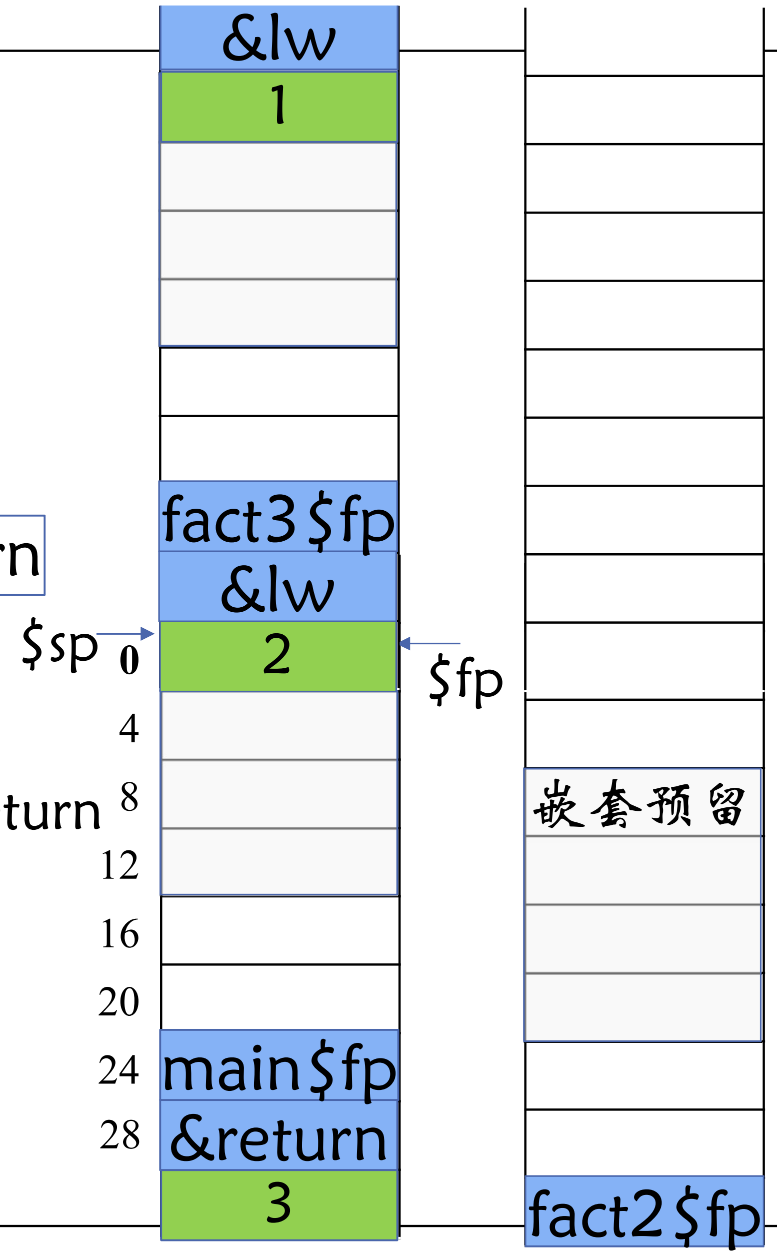
&lw

```
lw $a0,32($fp)
add $sp,$fp,0
lw $fp,24($sp)
lw $ra,28($sp)
addi $sp,$sp,32
mult $v0,$a0
mflo $v0
jr $ra
```

fact(3)返回

```
$v0=2
$a0=3
$ra=&return
$v0=6
```

返回到main&return



小结

- 递归子程序构成
- 递归程序执行时，栈的变化