

Non-Leaf Procedure Example

- C code:

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

- Argument n in \$a0
- Result in \$v0
- slt and slti instructions – compare two registers and sets a third register to 1

Examples: slt \$t0, \$s3, \$s4 # \$t0=1 if \$s3<\$s4

slti \$t0, \$s2, 10 # \$t0=1 if \$s2<10

Non-Leaf Procedure Example

- MIPS code:

fact:		
addi \$sp, \$sp, -8	# adjust stack for 2 items	
sw \$ra, 4(\$sp)	# save return address	
sw \$a0, 0(\$sp)	# save argument	
slti \$t0, \$a0, 1	# \$t0=1 if \$a0 < 1 (n<1)	
beq \$t0, \$zero, L1	# jump to L1 if \$t0=0(n>=1)	
addi \$v0, \$zero, 1	# if so, result is 1	
addi \$sp, \$sp, 8	# pop 2 items from stack*	
jr \$ra	# and return	
L1: addi \$a0, \$a0, -1	# else decrement n	
jal fact	# recursive call	
lw \$a0, 0(\$sp)	# restore original n	
lw \$ra, 4(\$sp)	# and return address	
addi \$sp, \$sp, 8	# pop 2 items from stack	
mul \$v0, \$a0, \$v0	# multiply to get result	
jr \$ra	# and return	

Note: \$a0 & \$ra do not change if n<1, so \$a0 & \$ra are not loaded before pop them

Non-Leaf Procedure Example: n=4 (1)

**Main call (\$a0)₁=4,
(\$ra)₁=return addr in main**

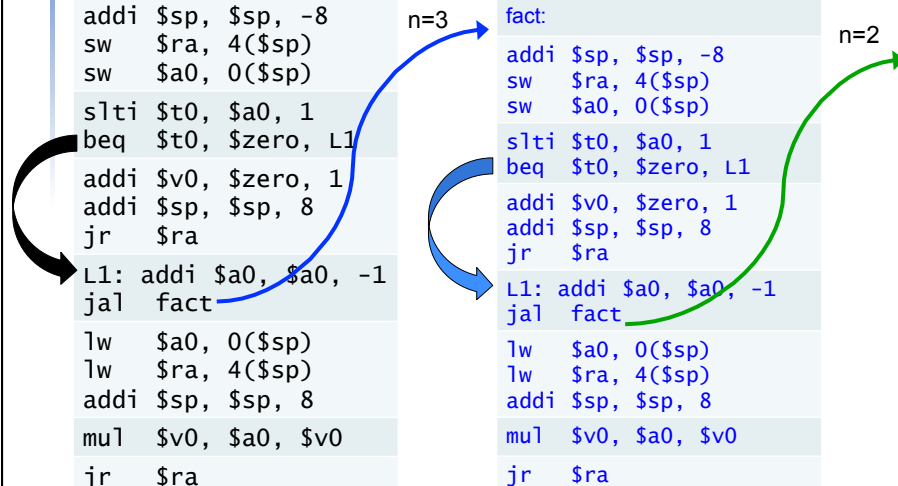
fact:

```
addi $sp, $sp, -8
sw  $ra, 4($sp)
sw  $a0, 0($sp)
slti $t0, $a0, 1
beq  $t0, $zero, L1
addi $v0, $zero, 1
addi $sp, $sp, 8
jr   $ra
L1: addi $a0, $a0, -1
jal  fact
lw   $a0, 0($sp)
lw   $ra, 4($sp)
addi $sp, $sp, 8
mul  $v0, $a0, $v0
jr   $ra
```

**fact1 call (\$a0)₂=3
(\$ra)₂=return addr in fact1**

fact:

```
addi $sp, $sp, -8
sw  $ra, 4($sp)
sw  $a0, 0($sp)
slti $t0, $a0, 1
beq  $t0, $zero, L1
addi $v0, $zero, 1
addi $sp, $sp, 8
jr   $ra
L1: addi $a0, $a0, -1
jal  fact
lw   $a0, 0($sp)
lw   $ra, 4($sp)
addi $sp, $sp, 8
mul  $v0, $a0, $v0
jr   $ra
```



Non-Leaf Procedure Example: n=4 (2)

**fact2 call (\$a0)₃=2,
(\$ra)₃=return addr in fact2**

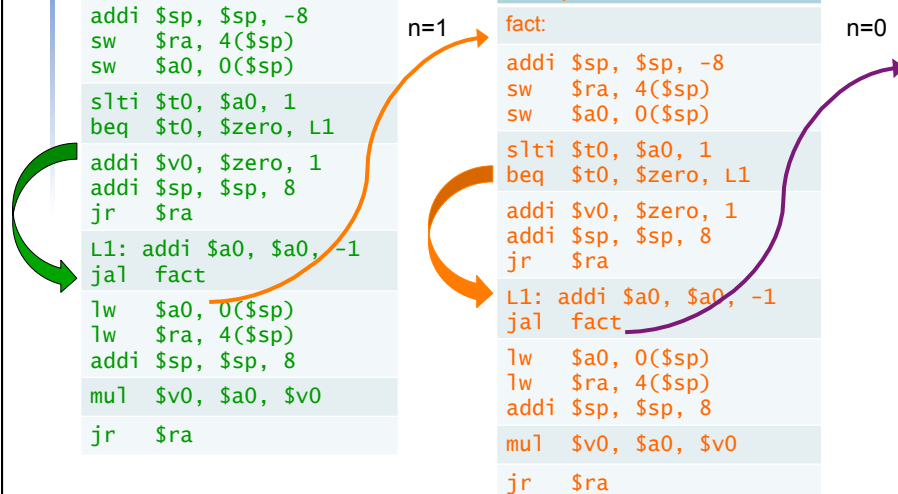
fact:

```
addi $sp, $sp, -8
sw  $ra, 4($sp)
sw  $a0, 0($sp)
slti $t0, $a0, 1
beq  $t0, $zero, L1
addi $v0, $zero, 1
addi $sp, $sp, 8
jr   $ra
L1: addi $a0, $a0, -1
jal  fact
lw   $a0, 0($sp)
lw   $ra, 4($sp)
addi $sp, $sp, 8
mul  $v0, $a0, $v0
jr   $ra
```

**fact3 call (\$a0)₄=1
(\$ra)₄=return addr in fact3**

fact:

```
addi $sp, $sp, -8
sw  $ra, 4($sp)
sw  $a0, 0($sp)
slti $t0, $a0, 1
beq  $t0, $zero, L1
addi $v0, $zero, 1
addi $sp, $sp, 8
jr   $ra
L1: addi $a0, $a0, -1
jal  fact
lw   $a0, 0($sp)
lw   $ra, 4($sp)
addi $sp, $sp, 8
mul  $v0, $a0, $v0
jr   $ra
```



Non-Leaf Procedure Example: n=4 (3)

fact4 call (\$a0)₅=0,
(\$ra)₅=return addr in fact4

fact:

```
addi $sp, $sp, -8
sw   $ra, 4($sp)
sw   $a0, 0($sp)
slti $t0, $a0, 1
beq  $t0, $zero, L1
addi $v0, $zero, 1
addi $sp, $sp, 8
jr   $ra
L1: addi $a0, $a0, -1
jal  fact
lw   $a0, 0($sp)
lw   $ra, 4($sp)
addi $sp, $sp, 8
mul  $v0, $a0, $v0
jr   $ra
```

fact3 call (\$a0)₄=1
(\$ra)₄=return addr in fact3

fact:

```
addi $sp, $sp, -8
sw   $ra, 4($sp)
sw   $a0, 0($sp)
slti $t0, $a0, 1
beq  $t0, $zero, L1
addi $v0, $zero, 1
addi $sp, $sp, 8
jr   $ra
L1: addi $a0, $a0, -1
jal  fact
lw   $a0, 0($sp)
lw   $ra, 4($sp)
addi $sp, $sp, 8
mul  $v0, $a0, $v0
jr   $ra
```

$\$v0=1$

$\$v0=1*($a0)_4$
 $=1$

Non-Leaf Procedure Example: n=4 (2)

fact2 call (\$a0)₃=2,
(\$ra)₃=return addr in fact2

fact:

```
addi $sp, $sp, -8
sw   $ra, 4($sp)
sw   $a0, 0($sp)
slti $t0, $a0, 1
beq  $t0, $zero, L1
addi $v0, $zero, 1
addi $sp, $sp, 8
jr   $ra
L1: addi $a0, $a0, -1
jal  fact
lw   $a0, 0($sp)
lw   $ra, 4($sp)
addi $sp, $sp, 8
mul  $v0, $a0, $v0
jr   $ra
```

$\$v0=1$

$\$v0=1*($a0)_3$
 $=2$

fact1 call (\$a0)₂=3
(\$ra)₂=return addr in fact1

fact:

```
addi $sp, $sp, -8
sw   $ra, 4($sp)
sw   $a0, 0($sp)
slti $t0, $a0, 1
beq  $t0, $zero, L1
addi $v0, $zero, 1
addi $sp, $sp, 8
jr   $ra
L1: addi $a0, $a0, -1
jal  fact
lw   $a0, 0($sp)
lw   $ra, 4($sp)
addi $sp, $sp, 8
mul  $v0, $a0, $v0
jr   $ra
```

$\$v0=2*($a0)_2$
 $=6$

Non-Leaf Procedure Example: n=4 (1)

\$v0=4

Main call (\$a0)₁=4,
(\$ra)₁=return addr in main

fact:

addi \$sp, \$sp, -8

sw \$ra, 4(\$sp)

sw \$a0, 0(\$sp)

slti \$t0, \$a0, 1

beq \$t0, \$zero, L1

addi \$v0, \$zero, 1

addi \$sp, \$sp, 8

jr \$ra

L1: addi \$a0, \$a0, -1

jal fact

lw \$a0, 0(\$sp)

lw \$ra, 4(\$sp)

addi \$sp, \$sp, 8

mul \$v0, \$a0, \$v0 # \$v0=6*(\$a0)₁=24

jr \$ra