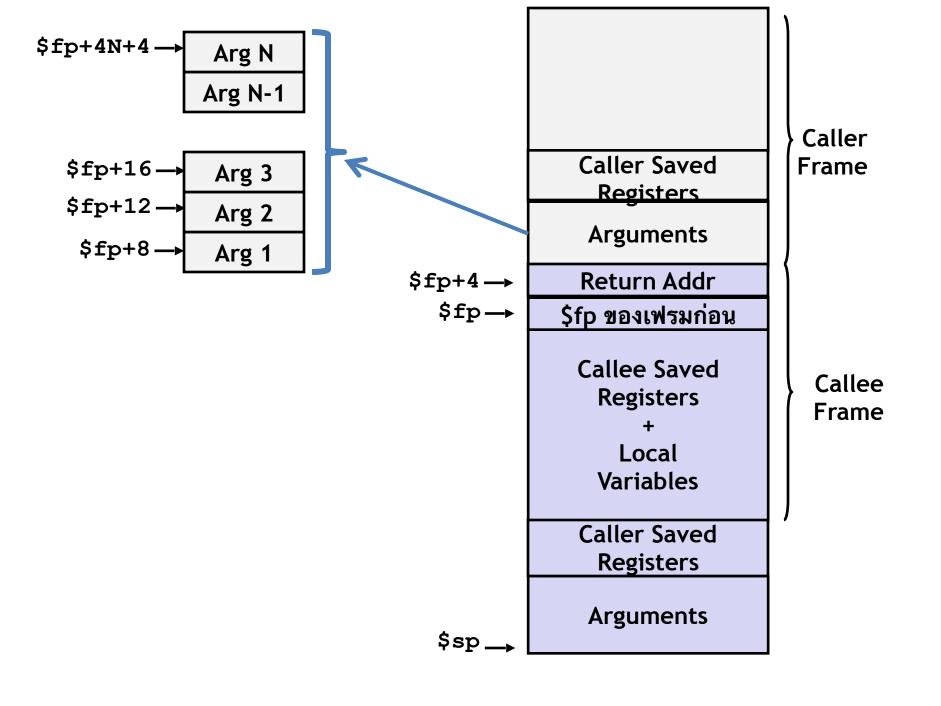
ข้อตกลงการใช้ Stack และ Recursion

ผู้สอน: ภารุจ รัตนวรพันธุ์

อะไรคือ Stack

- ส่วนของ memory ที่ใช้เก็บข้อมูลที่จำเป็นเวลา function/procedure ถูกเรียกใช้งาน
- แต่ละ function/procedure มี frame ของตัวเอง
- ขอบเขตของ frame กำหนดด้วย frame pointer กับ stack pointer
- Caller (โปรแกรมส่วนที่เรียกใช้งาน function)
 - save ค่า register ที่อยู่ใน set caller save
 - ส่งผ่าน arguments ลงบน stack
- Callee (function ที่ถูกเรียกใช้งาน)
 - save ค่า return address ลงบน stack
 - เก็บค่า frame pointer
 - Save ค่า register ที่อยู่ใน callee save
 - จองที่สำหรับ local variables



Factorial ในภาษาซี

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

L1: fact: addi \$sp, \$sp, -8 lw \$ra, 4(\$fp) sw \$ra, 4(\$sp) lw \$fp, 0(\$fp) sw \$fp, 0(\$sp) addi \$sp, \$sp, 8 move \$fp, \$sp **Factorial** addi \$v0, \$0, 1 lw \$t0, 8(\$fp) ใน MIPS slti \$t1, \$t0, 1 jr \$ra **Assembly** bne \$t1, \$0, L1 addi \$t2, \$t0, -1 main: addi \$sp, \$sp, -4 sw \$t2, 0(\$sp) addi \$sp, \$sp, -4 jal fact Termination case addi \$sp, \$sp, 4 sw \$t3, 0(\$sp) lw \$t0, 8(\$fp) jal fact mul \$v0, \$v0, \$t0 addi \$sp, \$sp, 4 lw \$ra, 4(\$fp) lw \$fp, 0(\$fp) addi \$sp, \$sp, 8 jr \$ra

Factorial ใน MIPS Assembly

fact:

Recursive case

addi \$sp, \$sp, -8 sw \$ra, 4(\$sp) sw \$fp, 0(\$sp) move \$fp, \$sp lw \$t0, 8(\$fp) slti \$t1, \$t0, 1 bne \$t1, \$0, L1 addi \$t2, \$t0, -1 addi \$sp, \$sp, -4 sw \$t2, 0(\$sp) jal fact addi \$sp, \$sp, 4 lw \$t0, 8(\$fp) mul \$v0, \$v0, \$t0 lw \$ra, 4(\$fp) lw \$fp, 0(\$fp) addi \$sp, \$sp, 8 jr \$ra

L1:

lw \$ra, 4(\$fp)
lw \$fp, 0(\$fp)
addi \$sp, \$sp, 8
addi \$v0, \$0, 1
jr \$ra

main:

addi \$sp, \$sp, -4 sw \$t3, 0(\$sp) jal fact addi \$sp, \$sp, 4

L1: fact: addi \$sp, \$sp, -8 lw \$ra, 4(\$fp) sw \$ra, 4(\$sp) lw \$fp, 0(\$fp) sw \$fp, 0(\$sp) addi \$sp, \$sp, 8 move \$fp, \$sp **Factorial** addi \$v0, \$0, 1 lw \$t0, 8(\$fp) ใน MIPS slti \$t1, \$t0, 1 jr \$ra **Assembly** bne \$t1, \$0, L1 addi \$t2, \$t0, -1 main: addi \$sp, \$sp, -4 sw \$t2, 0(\$sp) addi \$sp, \$sp, -4 jal fact sw \$t3, 0(\$sp) addi \$sp, \$sp, 4 Returning from lw \$t0, 8(\$fp) jal fact recursion mul \$v0, \$v0, \$t0 addi \$sp, Ssp, 4 lw \$ra, 4(\$fp) lw \$fp, 0(\$fp) addi \$sp, \$sp, 8 jr \$ra

การเติบโตของ stack สำหรับ fact(3) 3 A: \$fp (fact(3)) \$fp (main) 2 **B**: \$fp (fact(2)) \$fp (fact(3)) 1 B: \$fp (fact(1)) \$fp (fact(2)) 0 **B**: \$fp (fact(0))

\$fp (fact(1))

fact:

jr \$ra

L1: addi \$sp, \$sp, -8 lw \$ra, 4(\$fp) sw \$ra, 4(\$sp) lw \$fp, 0(\$fp) sw \$fp, 0(\$sp) addi \$sp, \$sp, 8 move \$fp, \$sp addi \$v0, \$0, 1 lw \$t0, 8(\$fp) slti \$t1, \$t0, 1 jr \$ra bne \$t1, \$0, L1 addi \$t2, \$t0, -1 main: addi \$sp, \$sp, -4 sw \$t2, 0(\$sp) addi \$sp, \$sp, -4 jal fact **B:** addi \$sp, \$sp, 4 sw \$t3, 0(\$sp) lw \$t0, 8(\$fp) jal fact mul \$v0, \$v0, \$t0 **A:** addi \$sp, \$sp, 4 lw \$ra, 4(\$fp) lw \$fp, 0(\$fp) addi \$sp, \$sp, 8