CS2318

10/11/16

REENTRANT

Reentrant Functions

No Global Variables

All Variables are stored on the stack

Consider Time Sharing

Round Robin on Users

Each Gets T seconds of Execution

(e.g. 10 ms)

Assume an editor program that

is supposed to serve numerous users

1) Make copies
2) Reentrant each user variables

are in Their personal stack

A recursive function

- Reentrant function that calls itself

Explicit Recursion

A:

JAL A:

jr $RA

Implicit Recursion

A calls B, C, D ... Z

Z calls A.

3

A: Root

JAL B:

exit

B: caller and callee

JAL C C - leaf

JR $RA

4) B: As a caller

B prepares A caller Frame

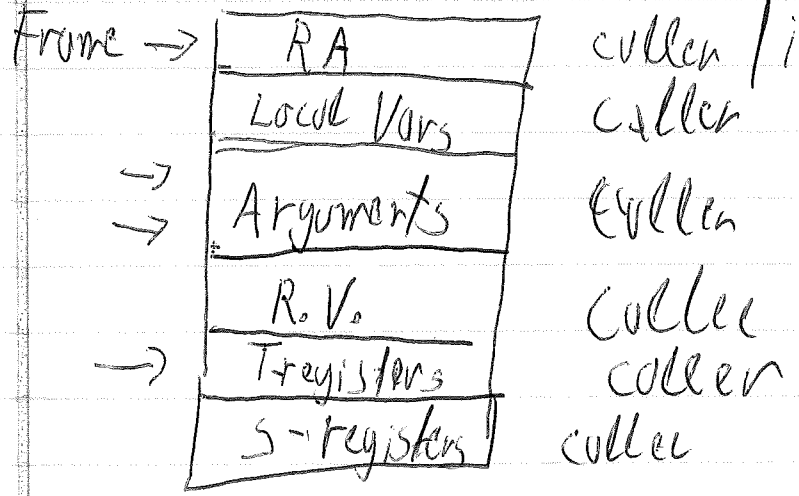
C: As a callee

Adds information to the frame

A Caller Frame A

B caller / callee Frame A
Frame B

C Working with frame B



5

A:

A makes a caller frame F_A

JAL B

~~B makes A call~~

exit

B:

Make F_B

Work as a callee on F_A

JAL C

JR $RA

C work as a

L:

callee on F_B

JR $RA

6

Push/pop on entire Frame

A:

JAL B

← RA of B from A

Exit

B:

[check Termination condition
No JR $RA (to B)

JAL B

← RA B from B

JR $RA

Prepare

B: caller FB-caller.

Word with B on FB caller

JAL B Last call get Back to A

Prepare a new frame

JR $RA as a caller

7

Iterative Factorial

Iteration vs. recursion

Efficient — overhead

Easier to program and maintain

Iteration ⟶ Recursion

⟵

Use a stack — Recursion on Index Value

```
int factorial(int n) {
    if (n == 1)
    Return (1)

    Return (n * factorial(n-1))
    }
```

8

```
int fib (int n)

return f(n-1) + f(n-2);

if (n == 0) return (0)

if (n == 1) return 1
```