Seg Ter Qua Qui Sex Sáb Dom

06/07/23

Nathann Zini dos Reis 19.2.4007

b-

Fib_iter:

```
bne   $a2, $0 , else   # se (n!=0) vai para else
move  $v0, $a1         # resultado = b
jr    $ra              # retorno para chamador
else:
addiu $sp, $sp, -4     # aloca frame = 4bytes
sw    $ra, 0($sp)      # salva endereço de retorno
move  $t0, $a0
addu  $a0, $a0, $a1    # $a0 = a+b
move  $a1, $t0         # $a1 = a
addiu $a2, $a2, -1     # $a2 = n-1
jal   fib_iter         # chamada recursiva
lw    $ra, 0($sp)      # libera pilha de frame
addiu $sp, $sp, 4      # retorno para chamador
jr    $ra
```

- Total de instruções de construção = n*11 + 3
- 11 inst. p/ cada chamada/retorno recursivo (se n>0)
- +3 inst. se (n==0)

1 1

Seg Ter Qua Qui Sex Sáb Dom

Nathann Zini dos Reis

06/07/23 Tarefa 3 19.2.4007

1- C → MIPS assembly

a-

Compare:

```
addi $sp, $sp, -4      # frame alocado = 4 bytes
sw   $ra, 0($sp)       # salvar endereço de retorno
jal  sub               # chama sub
li   $t0, 0            # resultado = 0
bltz $v0, exit         # se sub(a,b) < 0 ir para exit
li   $t0, 1            # resultado = 1
exit:
move $v0, $t0          # $v0 = resultado
lw   $ra, 0($sp)       # restaura endereço de retorno
addi $sp, $sp, 4       # libera pilha de frame
jr   $ra               # retorna para chamador
sub:
sub $v0, $a0, $a1      # resultado = a-b
jr $ra.                # retorna para o chamador
```

- 11 ou 12 instruções (dependendo se o bltz é levado ou não). Inclui a chamada e retorno de sub.

FÓRONI