

ROTINAS E PILHA

chamada e retorno de rotinas

- pil a_2 , L → quando em a_2 o endereço da próxima instrução é puxa para o rótulo L
↳ $PC + 4 = ra$ (return address)
- ret → salta imediatamente para a função ra
- problema → quando uma rotina chama uma subrotina, perdemos o endereço de ra!
 - solução → salvar o endereço de retorno antes de invocar uma rotina!
↳ SALVAR NA PILHA DO PROGRAMA!

pilha de programa

- rotina ativa → rotina que não retorna → necessidade de armazenar seus dados
 - antes de retornar, o código da própria rotina desaloca o espaço alocado na pilha
 - cresce para baixo → começo da pilha de memória maior e vai diminuindo
↳ a última posição preservada na pilha
- SP → Stack Pointer → aponta para o topo da pilha
 - empilhar o valor de a_0 →

addi op, op, -4
sw a_0, 0(op)

 → desaloca, e negativo pq que cresce para baixo
→ alocar o valor
 - desempilhar o valor de a_0 →

lw a_1, 0(op)
addi op, op, 4

 → lê o valor
→ desaloca, sp volta a apontar pra o topo anterior
 - empilhando vários valores →

addi op, op, -12
sw a_0, 8(op)
sw a_1, 4(op)
sw a_2, 0(op)
 - desempilhando vários valores →

lw ...
addi op, op, 12

 → quantidade de valores alocados!
- ATENÇÃO! → último empilhado deve ser o 1º desempilhado!

salvando o endereço de retorno

→ "tudo que empilha, desempilha no final" →

funções:
addi op, op, -4
sw ra, 0(op)
outra função
lw ra, 0(op)
addi op, op, 4

→ empilha → endereço de retorno
→ adiciona ra na pilha
→ chama outra função
→ carrega ra, volta o endereço
→ desaloca / desempilha

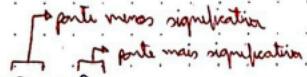
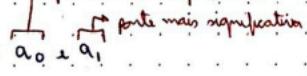
ABI E A POLÍTICA DE USO DOS REGISTRADORES

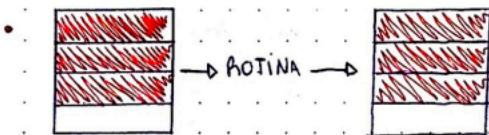
comunicação de chamada

- define onde os parâmetros de uma função devem ser passados

- permite o uso global de funções!

retorno de valores das funções

- até 32 bits $\rightarrow a_0$  parte menos significativa
- entre 32 e 64 bits $\rightarrow a_0 \dots a_1$  parte mais significativa
- próximos $\rightarrow a_0, a_1, \dots, a_7$, sendo que o resto vai pela pilha!
• antes da chamada, os parâmetros são empilhados \rightarrow depois de executar a rotina, é desempilhado



- obs \rightarrow empilha 1º o último, pois minha função deve dar os valores na ordem correta
parâmetro de parâmetros por valor

- li a_0 , valor

parâmetro de parâmetros por referência

- antes da função \rightarrow [la a_0 , endereço do valor]

- dentro da função \rightarrow [la t_1 , 0(a0)
:
ou t_1 , 0(a0)]

política de uso de registradores

- calle - varid \rightarrow outros registradores \rightarrow registradores vazios pela rotina chamada

- calle - varid \rightarrow só a $r_11 \rightarrow$ registradores vazios pela rotina volta

quadros de pilha \rightarrow rotinas altas não adicionando quadros para suprir suas necessidades

- frame pointer \rightarrow aponta para o início do quadro atual \rightarrow 