

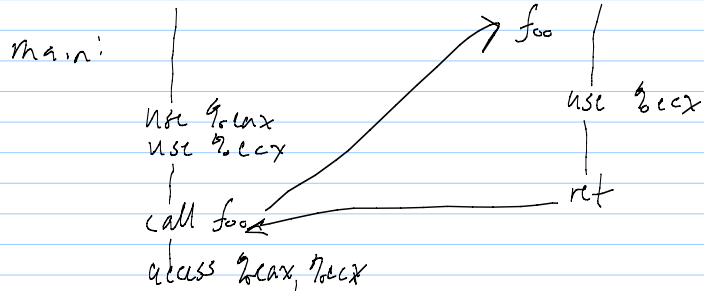
Call/Return Protocol - Stack Frame

Note Title

10/17/2007

observe: registers used for efficient access to temp./intermediate data

issue: registers are global, software dev. is local



one bad solution

every subroutine documents its reg. usage

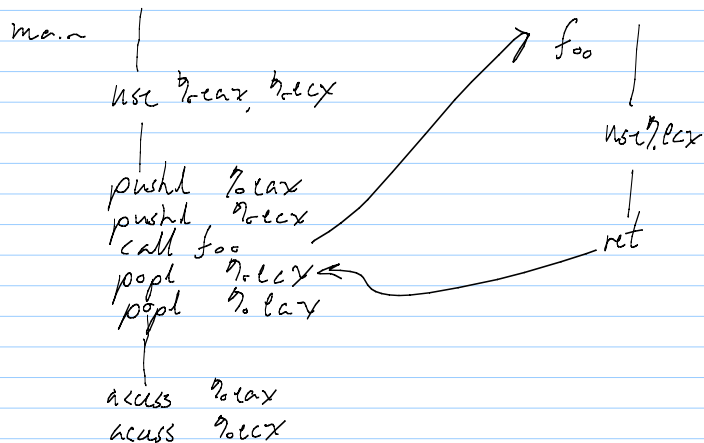
#sub * #calls = #way to goot

⇒ save registers if to be used in caller routine and caller routine

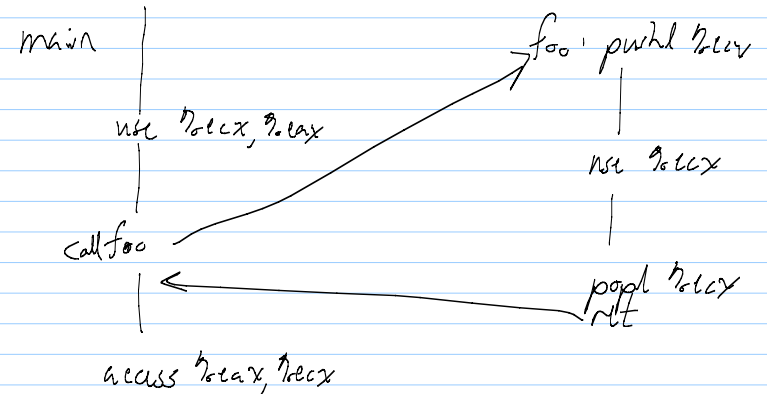
stack is trad. used for this purpose

who is responsible?

caller save registers



callee save

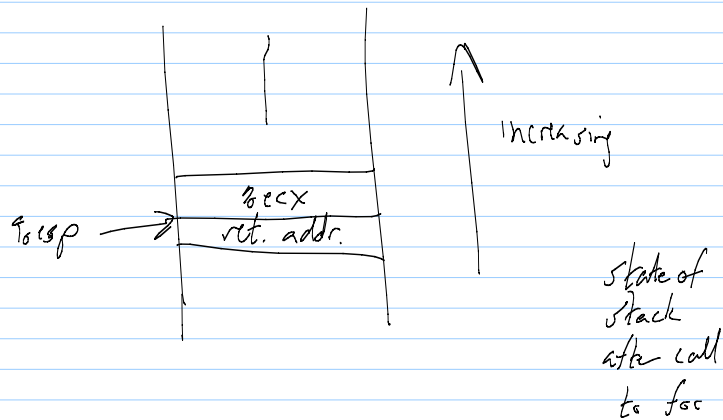
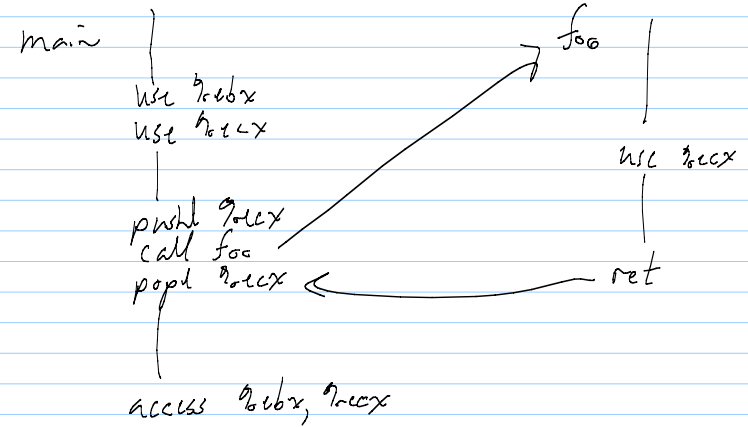


mixture of callee and caller save

caller save $\%eax$, $\%ecx$, and $\%edx$
callee save $\%ebx$, $\%esi$, and $\%edi$

perform save only when needed based on local info.

move $\%eax$ to $\%ebx$



other uses for stack

- passing parameters to subroutines
- local variables within "

⇒ motivates stack frame

region of stack assoc. w/
particular invocation of func./subroutine

top of stack always available via $\%esp$
stable reference relative to call $\%ebp$
base pointer frame pointer

