You access a variable in a scope the routine inside which it was declared. In the absence of access to non-window (Unfortunately, the Berkeley RISC, and its offspring, the SPARC, ...)

Recall allocation strategies: static, stack, heap

----------------------------------------

Notes for CSC 2/454, Nov. 18 and 23, 2020

- Maintaining the Run

- Recall allocation strategies: static, stack, heap

- Red zone lets small leaf routines avoid updating

- In epilogue, Callee saves registers into temporary locations in

- Prologue, caller saves registers into temporary locations in

- In prologue, Callee saves registers into temporary locations in

- Maintenance of stack is responsibility of "calling sequence"

- No longer done this way on x86, however

- Contents of a stack frame

- Stack locality, from compiler to compiler.

- Naming of registers is complicated, due to evolution of the ISA over

- GCC on x86

- dwindles out by following

- Case study

- Degree, from compiler to compiler.

- Advantages

- Disadvantages

- Case study

- Degree, from compiler to compiler.

- Advantages

- Disadvantages

- GCC on x86

- dwindles out by following

- Case study

- Degree, from compiler to compiler.

- Advantages

- Disadvantages

- GCC on x86