Lecture Notes 14

Implementing "Simple" Subprograms

- Semantics to call "simple" subprogram
  1. Save execution status of current program unit
  2. Compute pass parameters
  3. Pass return address to called subprogram
  4. Transfer control to called subprogram
- Semantics to return from "simple" subprogram
  1. Copy out any out-mode/pass-by-value-result values
  2. Move result (if function) to location accessible to caller
  3. Restore execution status to caller
  4. Transfer control to caller
- Activation Record – Format of the noncode part of subprogram
- Activation Record Instance – Concrete example of activation record

Implementing Subprograms with Stack-Dynamic Local Variables

- Semantics to call subprogram
  1. Create Activation Record
  2. Save execution status of current program unit
  3. Compute pass parameters
  4. Pass return address to called subprogram
  5. Transfer control to called subprogram
- Prologue actions of called
  1. Save old EP in the stack as dynamic link and create new one
  2. Allocate local variables
- Semantics to return from subprogram
  1. Copy out any out-mode/pass-by-value-result values
  2. Move result (if function) to location accessible to caller
  3. Restore stack pointer by setting to EP
  4. Restore execution status to caller
  5. Transfer control to caller
- Call Chain (or Dynamic Chain) – Collection of dynamic links

Nested Subprograms & Blocks

- Static Link – A part of activation record that points to the bottom of the activation record
- Static Chain – A chain of static links
- Static Depth – How deeply nested a subprogram is from the outermost scope

- Nesting Depth (or Chain Offset) – Difference in Static Depth from non-local variable scope and the Static Depth of the subprogram referencing it
- Block Activation Record – Entry/exit locations are strictly known so allocation can be statically determined

Implementing Dynamic Scoping

- Deep Access – Following dynamic links to find non-local reference in activation record
- Shallow Access – A separate stack is maintained for each variable