

Programming Languages

Storage management

Storage Management

- Consider the ways in which various languages arrange to have space made available to the object program
- and how it may be arranged to have that space used economically

Static storage allocation

- If the size of every data item can be determined by the compiler
- If recursive procedure calls are not permitted
- The space for all programs and data can be allocated at compile time – statically
- Easy to implement, does not need any runtime-support
- E.g., FORTRAN
 - The space required is the sum of the space needed for
 - Subprograms
 - Data
 - Linkage information of return address
 - Any library routines used
 - Space does not change while running

Dynamic Storage Allocation

- If recursive procedure, adjustable arrays are permitted then
- Dynamic storage allocation is required
- Two kinds
 - Stack allocation
 - Heap allocation
 - Useful for implementing data whose sizes varies at run-time

Stack allocation of storage

- Allocation of storage will be done in the form of a stack
 - Storage addition done on the top of the stack and
 - Releasing of storage also done by removing from the top of the stack
 - Decrementing the stack pointer
- All the fixed size storage required by variables that has been declared in a procedure into a single chunk of storage called activation record
- Contents of an activation record
 1. Storage for simple names, pointers to arrays and other data structures local to the procedure
 2. Temporaries for expression evaluation and parameter passing
 3. Information regarding attributes for local names and formal parameter when they cannot be determined at compile time
 4. The return address
 5. A pointer to the activation record of the caller

Stack allocation of storage

- P calls Q
- The activation record for Q is placed on top of the stack
- When Q returns, the return address is fetched from the activation record
- Activation record of Q is removed from the stack by decrementing stack pointer

Recursion and Displays

- When a language has recursive procedures
- Then several activation record of the same block/procedure appear on the stack
- Method used in ALGOL is the **display**
- In addition to the stack there is a **list of pointers**- called a **display**
- The display has a pointer to the correct activation record for every procedure in the environment of the currently active procedure

Stack allocation with dynamic binding

– SNOBOL

Heap allocation

- For data whose sizes fluctuates, it is inconvenient to place them on the stack
- A very useful run-time organization is the **heap**
 - A large block of storage that can be partitioned into smaller blocks
 - The portion of the heap not currently in use are linked together in an **available space list**