Run-Time Storage Organization

66.648 Compiler Design Lecture (03/23/98)

Computer Science

Rensselaer Polytechnic

Lecture Outline

- Run-Time Storage Organization
- Examples
- Administration

Run-Time Storage Organization

A program obtains a single contiguous block of storage (virtual memory) from the operating system at the start of program execution. The generated code assumes a subdivision of the storage into different areas/

Code -- this area contains the generated target code for all procedures in the program. The size of this can be determined statically by the compiler/linker.

Static Data

Static Data-- this area contains global data objects whose size can be determined statically at compile time. Static variables are mapped to offsets in the static data area.

Stack

Stack--runtime stack of activation records reflecting the stack structure of dynamic procedure calls and returns. An activation record contains the information needed by a single procedure call. Local variables are mapped to offsets in the activation record.

Heap

Heap -- used to store all other program data (data that is dynamically sized or data with lifetime pattern that cannot be represented in the run-time stack). Heap data allocation incurs more overhead than static or stack data allocation.

Languages and Storage Organization

Fortran - Static

Pascal - Stack/heap

C Stack/Heap

Activation Record

```
Public class X {
public static void main(String argsv[]) {
int n;
n =10;
System.out.println(fib(n));
}
int fib(int n)  {
if (n==0) return 1 else if (n==1) return 1 else
    return(fib(n-1)+fib(n-2)); }}
```

Compile Time Layout of Local Data

Local variables offsets are calculated depending on its size. These offsets can be computed relative to the start of the stack frame, and can be used to specify the layout of local data in the stack frame.

Local variables accesses get translated to negative-valued offsets on the stack pointer.

Non-Local Variable

An access to a lexically scoped nonlocal variable gets translated to <level-count,frame-offset>.

Level-count = k indicates that the variable can be found in the kth enclosing scope.

Variable/Field of Unknown Size

How do we allocate storage for a variable/ field with statically unknown size e.g. an array or an adt (abstract Data Type).

By allocating storage separately (on the heap or on top of the AR on the stack) and storing a pointer to the storage in the AR.

Other Basic Concepts in Runtime Structues

Call Sequence -- instruction Sequences that allocates AR for callee procedures and fills in some of the fields.

Return Sequence- instruction sequence that restores machine state so that caller procedure can continue execution.

Lexical scope -- Nonlocal names are resolved via static nesting.

Dynamic Scope: nonlocal names are resolved by inhering name bindings (following control links)

Example

Parameter Passing

Call by value -- caller places r-value for actual parameter in the storage formal parameter.

Call by reference -- caller places l-value for actual parameter in the storage for formal parameter.

Call by value result -- caller places r-value for the actual parameter in the storage for formal parameter and also determines the l-value of the actual parameter. On return, the current r-value of the formal parameter is copied to the l-value of the actual parameter.

Example

Comments and Feedback

Project 3 is out. Please start working. PLEASE do not wait for the due date to come.

We are in chapter 8. Please read that chapter and read the relevant portion of Java. Please keep studying this material and work exercises.