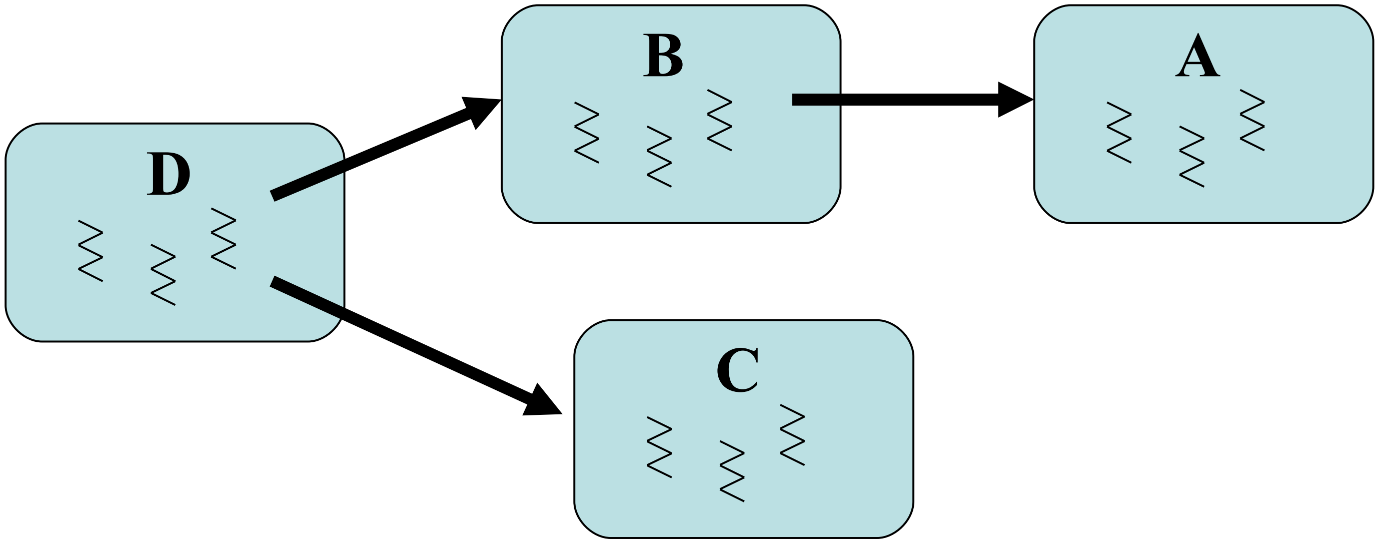
Composable Parallel Programming



Any parallel program can be used, without change, as a component of a larger parallel program: A can be used to build B; B and C to build D.

Composable Parallel Programming

Q: **Why is it not possible today?**

A: **Parallel Programs make Resource Decisions!**

Which Processors?
How is Data Distributed?

Q: **Why can’t a Compiler do it?**

A: **Must Look at the Entire Program!**
Can’t Respond to Run-Time Changes!

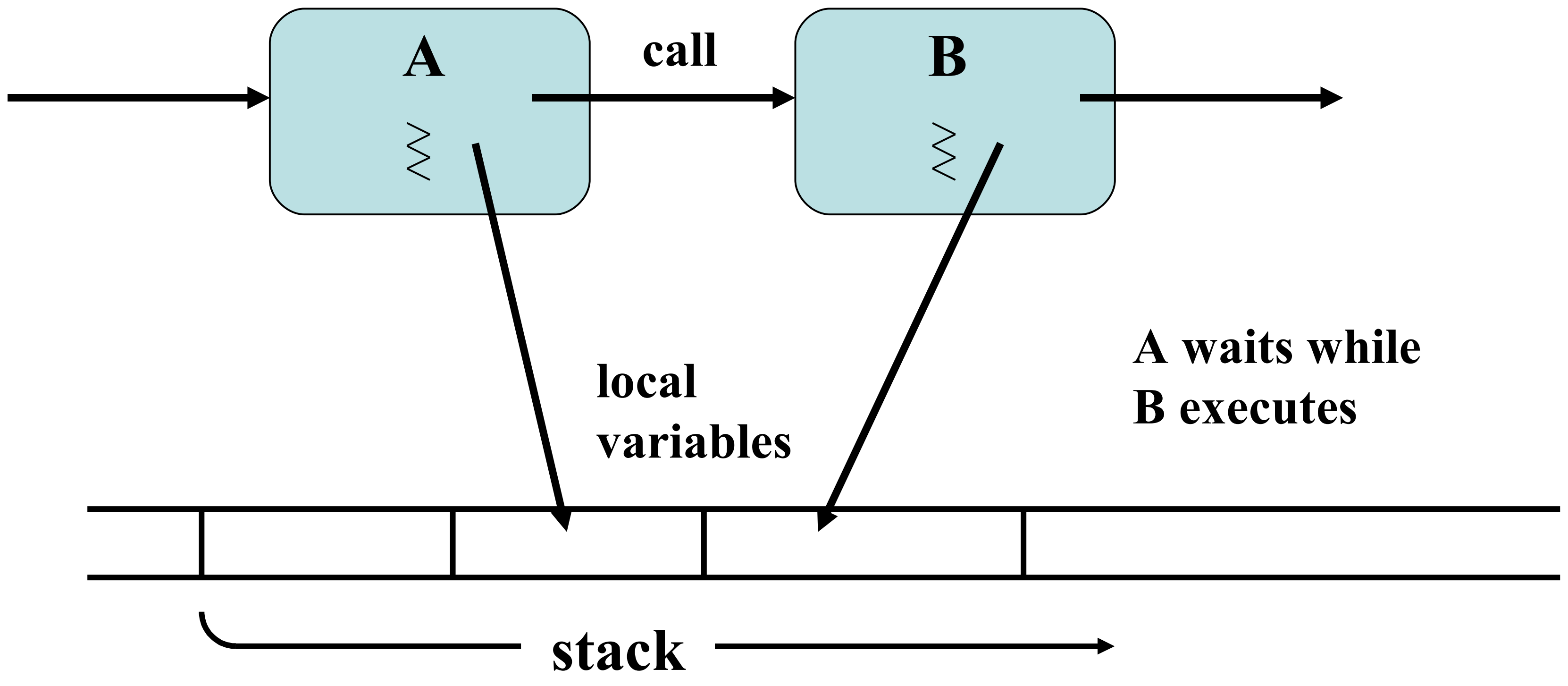
Q: What can we learn from Sequential Computing?

A: Two Powerful Concepts:

Procedures Objects

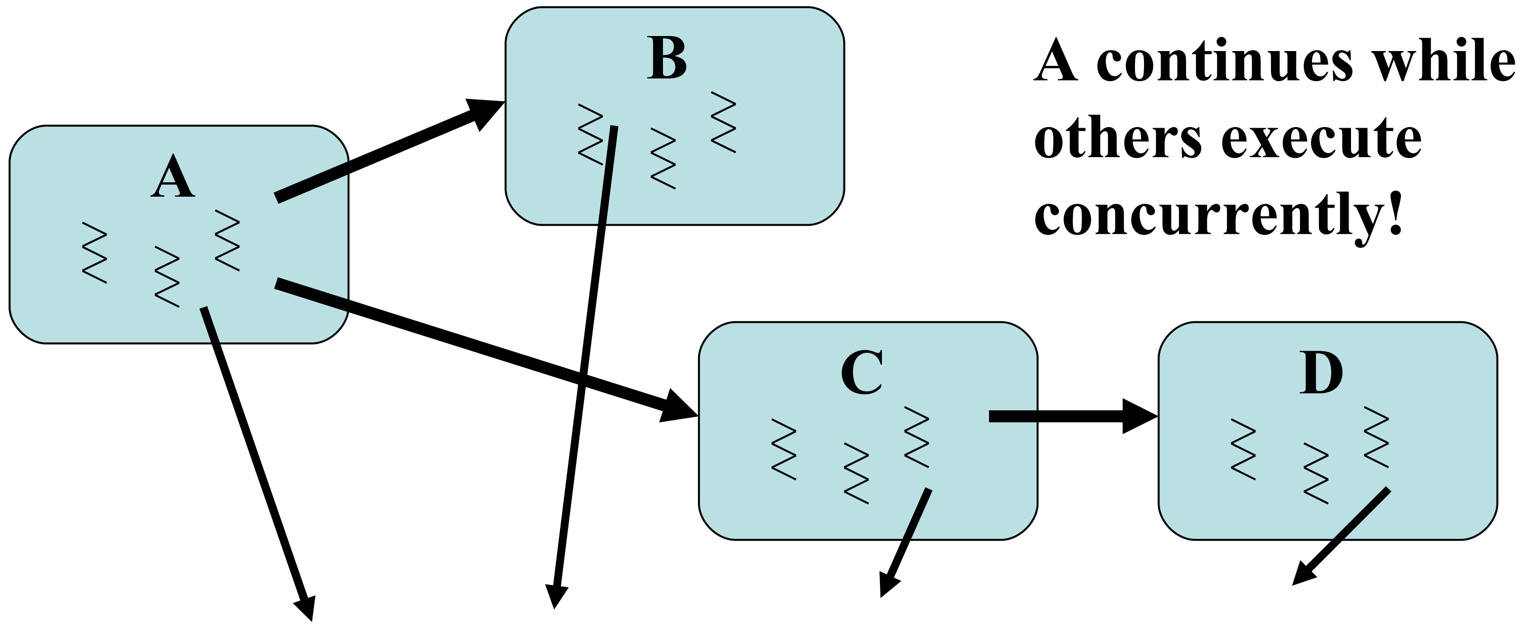
Goal: Generalize for Multi-Core Computing!

Nested Sequential Procedure Activations



Address Space of Program

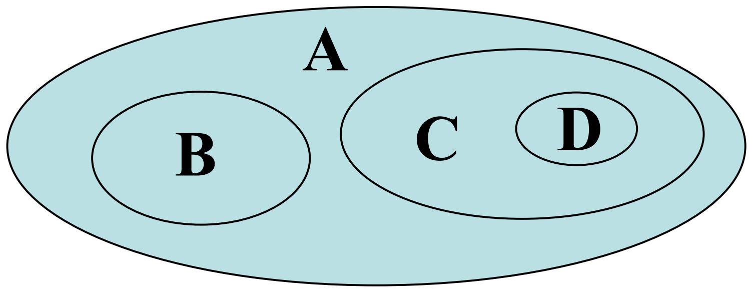
Nested Parallel Procedure Activations



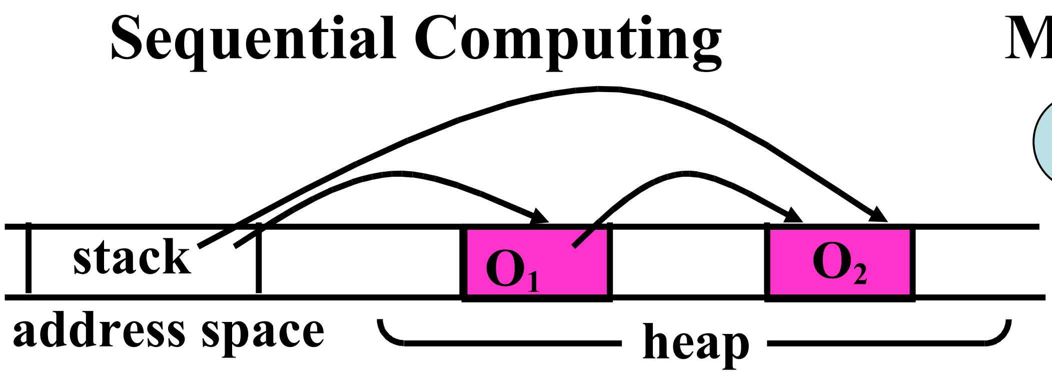
A continues while others execute concurrently!

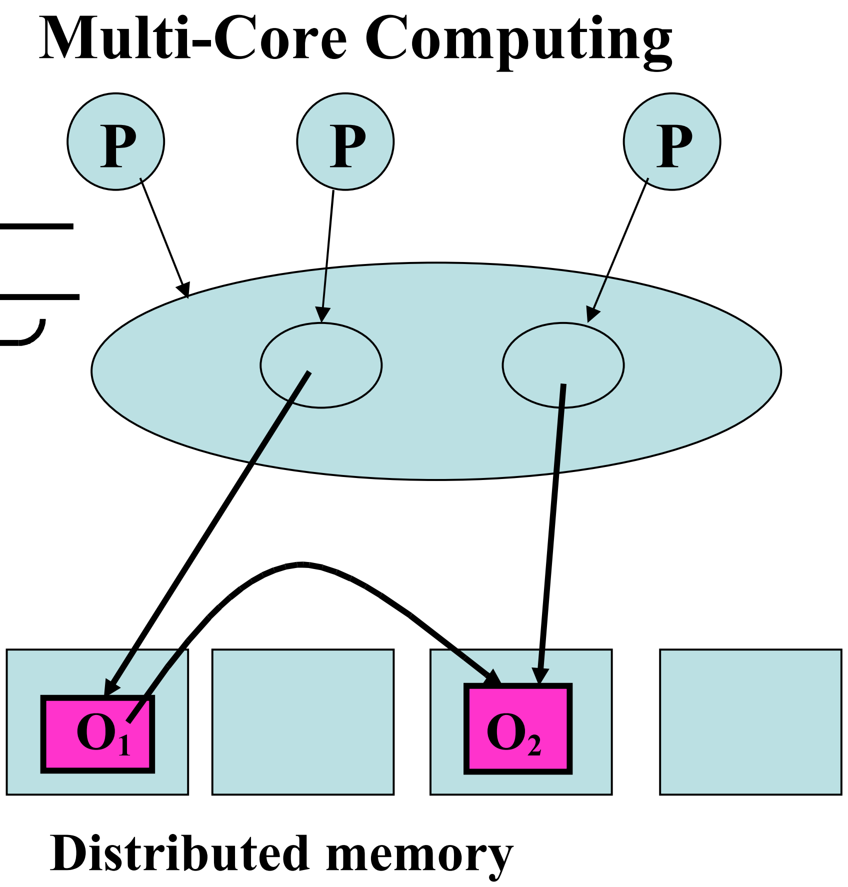
Q: How to Allocate Locals? A Stack no longer works!

A: Use a Tree Structure; Allocate in Heap
The Cactus Stack:



General Object Implementation





To avoid complex address translation all objects must reside in a common global address space.

Composable Parallel Programming

What it requires: **Efficient Heap Management**

Fine-Grain Scheduling of Threads

Q: **Can efficient Heap Management be Implemented in a Multi-Core Computer?**

A: **Yes! With the right hardware:**
- Global Addressing
- Built-in Garbage Collection

Composability

Current multiprocessor computers do not support composition of parallel programs:

Using a parallel program as a component of a larger parallel program generally requires understanding and modifying the internal mechanisms of the component.

This is true because programmers are given the responsibility for planning the management of processors and distribution of data.

Requirements for Composability

- Means for flexible and fast run-time management of processor and memory allocation.
  - Hardware-supported memory allocation and garbage collection.
  - Fine-grain scheduling of threads.
- Architectural support for a memory model that satisfies principles of modularity:
  - Context Independence
  - Data Generality
  - Hierarchy