Peer Instruction #8: Memory Model/Stack Convention

Which programming languages use an execution model based on a stack?

A. Java
B. C, C++
C. Pascal, Fortran
D. Algol, Ada, Prolog
E. All of the above

Execution Models

Why not discard the stack convention and just use register passing?

A. Without stack, recursion is impossible
B. No alternative memory models exist
C. Overall, no more better approach exists
D. Number of registers is too limited
E. Register passing is too complicated

Stack Convention

Which of the following is not stored on the stack?

A. Dynamic allocations
B. Function parameters
C. Return values
D. Local variables
E. Return addresses

Stack Convention

Does the caller or callee have to push function parameters?

A. Caller function
B. Callee function
C. Either, depending on the protocol

Stack Convention

Does the caller or callee have to pop the return value of a function?

A. Caller function
B. Callee function
C. Either, depending on the protocol

Stack Convention

Does the caller or callee have to allocate space for the return value?

A. Caller function
B. Callee function
C. Either, depending on the protocol

Stack Convention

Why is it necessary to have both a frame pointer and stack pointer?

A. Because the stack pointer is volatile, and constantly changes

B. Because the stack pointer is shared between all functions

C. To have a stable pointer for accessing function parameters and locals

D. All of the above

Frame Pointer

What does the following LC-3 assembly code do?

```
ADD R6,R6,#-1
STR R5,R6,#0
…
LDR R5,R6,#0
ADD R6,R6,#1
```

A. Initializes the frame and stack pointer
B. Pushes and pops the frame pointer
C. Pushes and pops the return address

Frame Pointer