Procedures and Functions

CMPSC 461
Programming Language Concepts
Penn State University
Fall 2016

2nd midterm:

Nov. 7, 6:30 to 7:45PM, 112 Kern Building

NOT cumulative, covers slides lec15 to lec26 (this lecture) and note3, note4 on Canvas

Practice problems are posted on Canvas

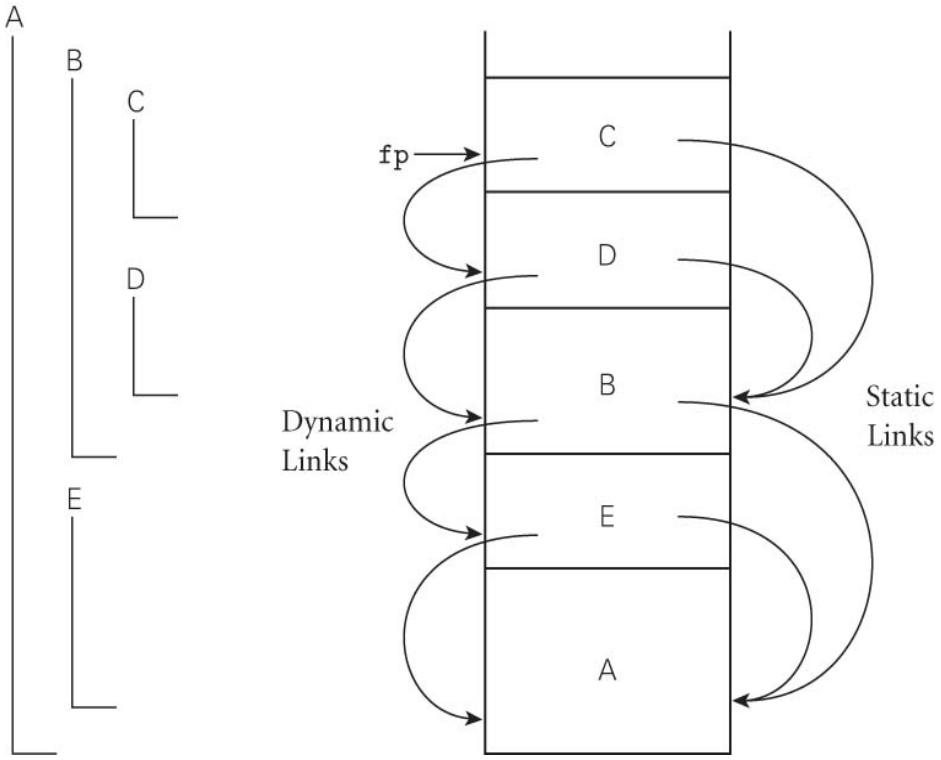
This Friday: review for midterm

Conflicts? ***Before this Friday (Nov. 4th)***:

Send Anindita (aib5487@psu.edu) and me an email, which

1. explains your conflict
2. proposes alternative time slots within the week of midterm

Locate Non-Local Variable



How can function C access a variable defined in A?
Follow SLs to find the frame of A

How many hops?
n(C)-n(A)=3-1=2

Locate Non-Local Variable

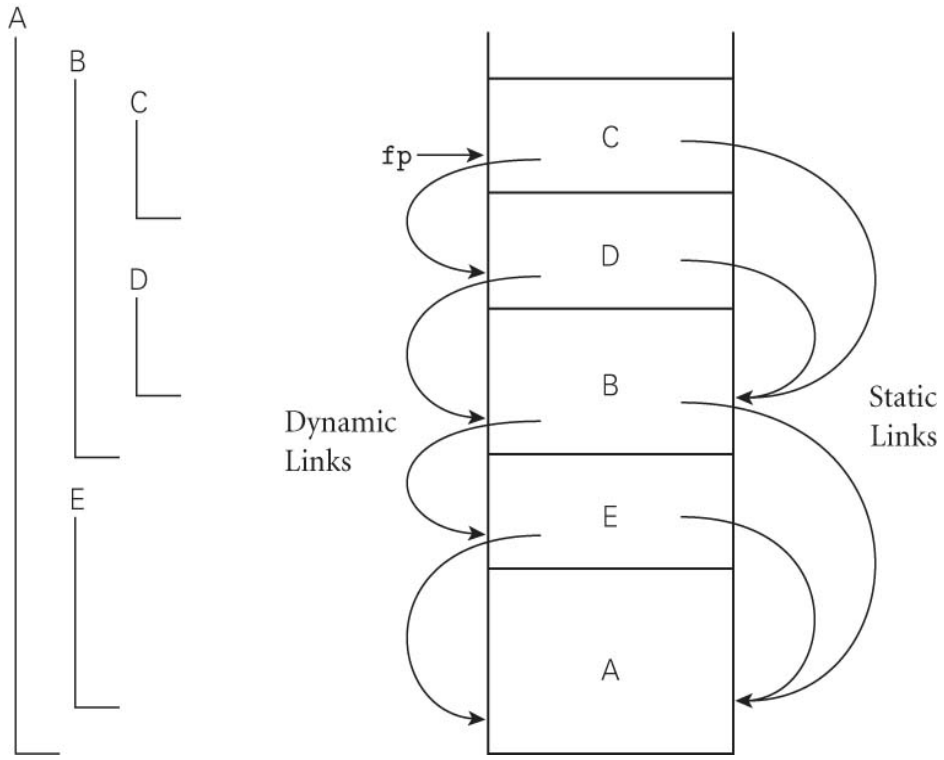
Resolve names using symbol tables

If current depth is k and the symbol table defining the variable has depth l, generate code:

- Compute base address: (k-l) hops following SL
- Find offset of variable
- Address at run time: base + offset

Can k be smaller than l?
The frame containing non-local variable must be on stack?

Static Links



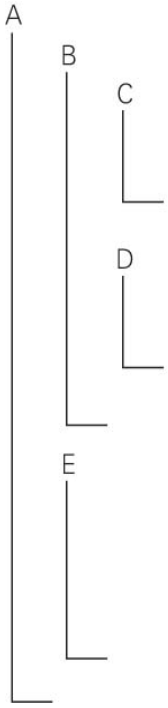
How are the static links maintained?

Nested Depth

n(p): nested depth of a function p

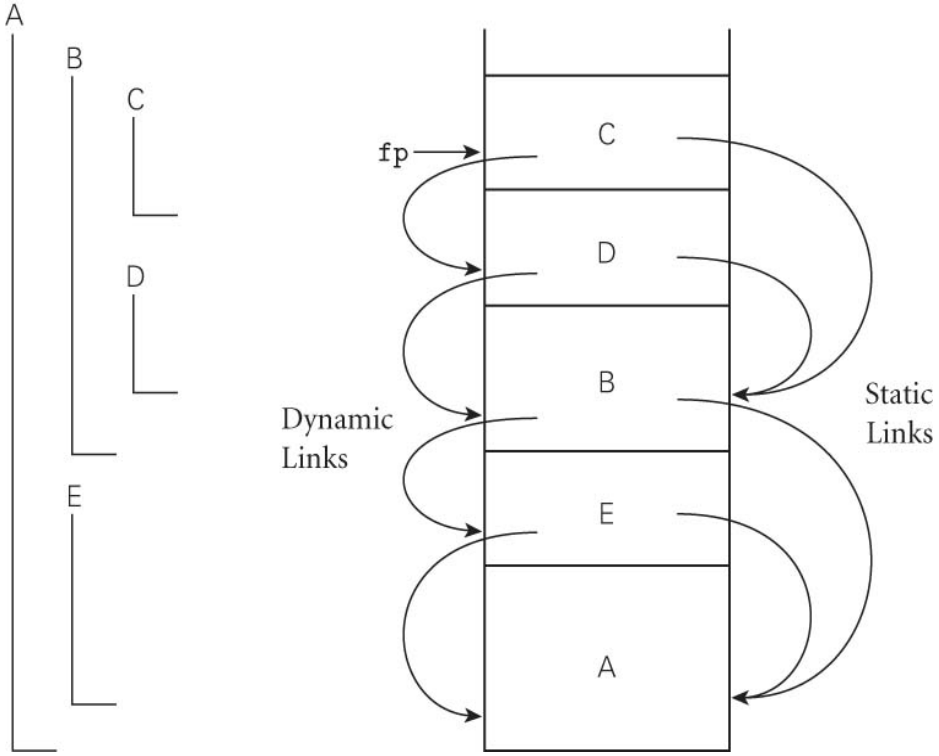
1 for functions not nested (top-level functions)

i+1 for functions immediately defined in depth i



n(A)=1
n(B)=2
n(C)=3
n(D)=3
n(E)=2

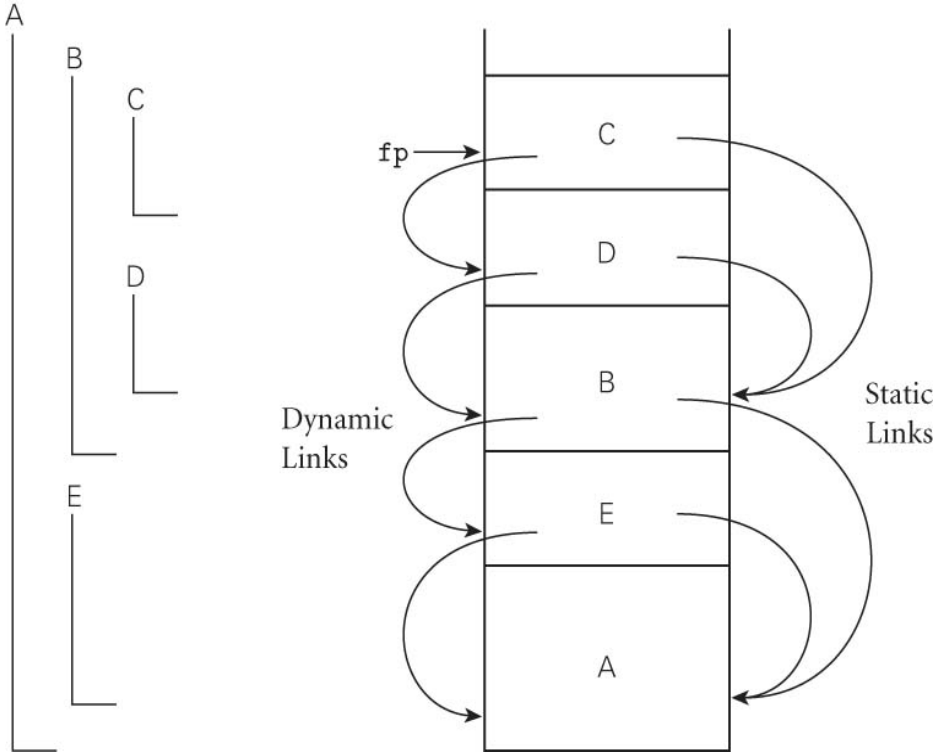
Static Links and Scoping: Callee is more nested



When function f1 calls f2 and n(f1)<n(f2) (callee Is more nested), *f1 must be the immediate lexical ancestor of f2*.

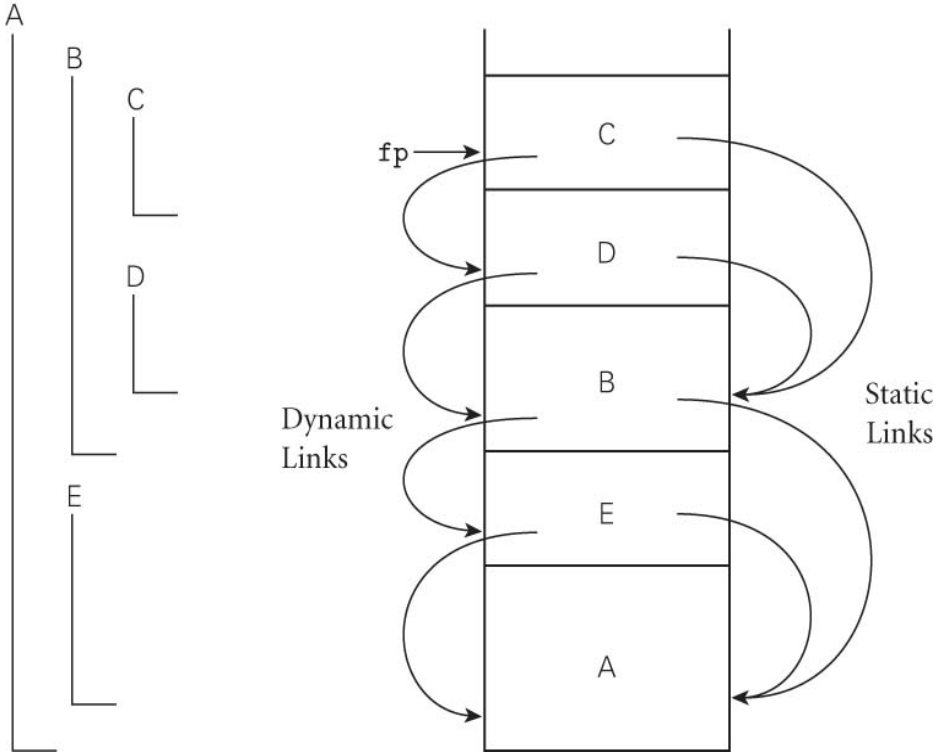
e.g., A calls E, B calls D

Static Links and Scoping: Caller is more nested



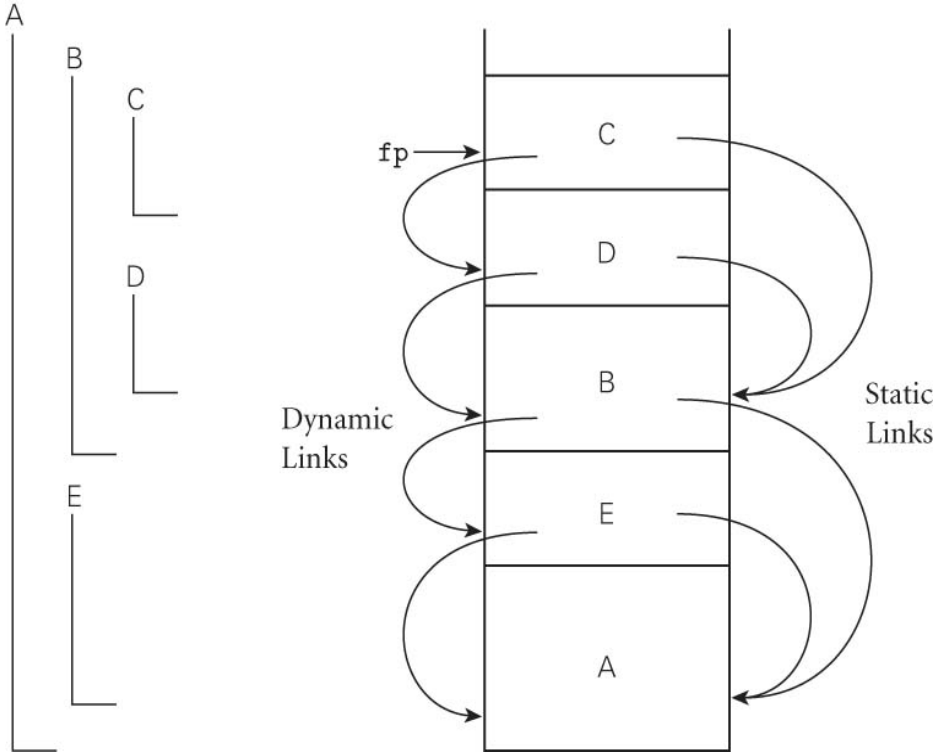
When function f1 calls f2 and n(f1) ≥ n(f2) (caller is more nested), f1 must be defined in some function f3, and f2 is immediately defined in f3.
e.g., E calls B, D calls C, D calls E

Manage Static Links: Callee is more nested



The static link for D when B calls D? (n(B)<n(D))
B must be the immediate lexical ancestor of D.
So SL is the same as DL.

Manage Static Links: Caller is more nested



The static link for C when D calls C? (n(D)≥n(C))
D must be defined in some function f, and C is immediately defined in f, where f being B. So Find AR of the closest common lexical ancestor of C and D, by following SLs from D for one (n(D)-n(C)+1) hop.

Manage Static Links

Explicit call (P calls Q)

- n(Q)>n(P): Q must be defined immediately in P. otherwise, not in scope

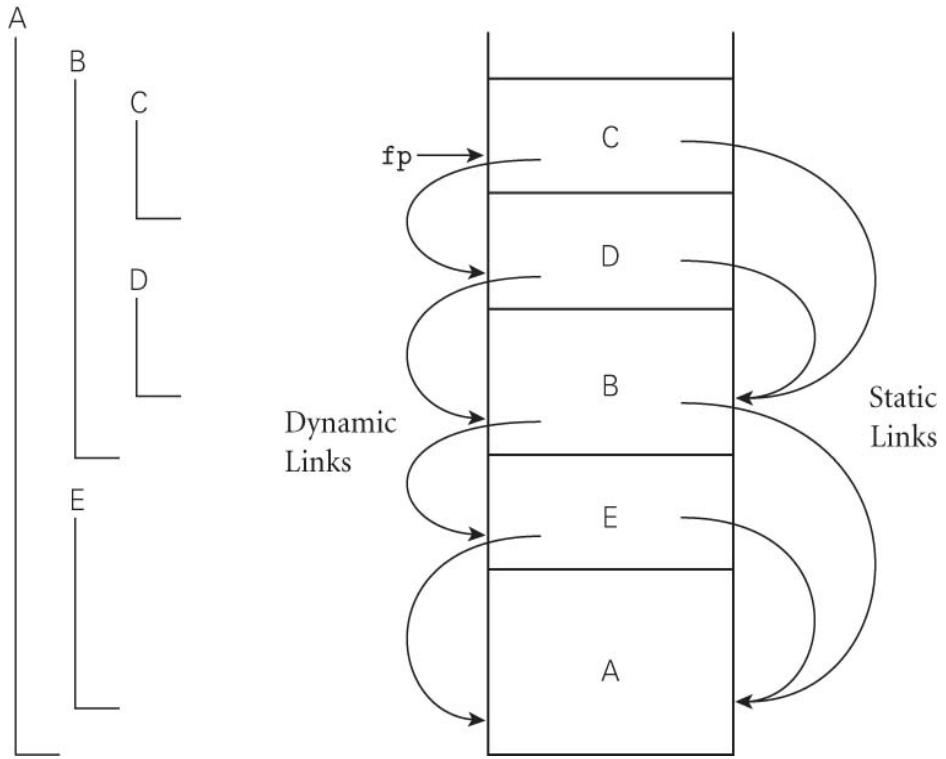
  Hence, put current frame pointer into the frame of Q

- n(Q)<=n(P): P must be defined in some function R, and Q is immediately defined in R

  Hence, follow static link of P for (n(P)-n(Q)+1) hops, put the last link into the frame of Q

  (special case: recursive call n(P)=n(Q))

Manage Static Links



What happens when D calls E instead of C?

Limitation of Static Links

High cost with highly nested program:

- (k-l) memory reads when P at depth k refers to a variable at depth l
- (n(P)-n(Q)+1) memory reads to establish SL when P calls Q

Display: embed static chain into an array

Display

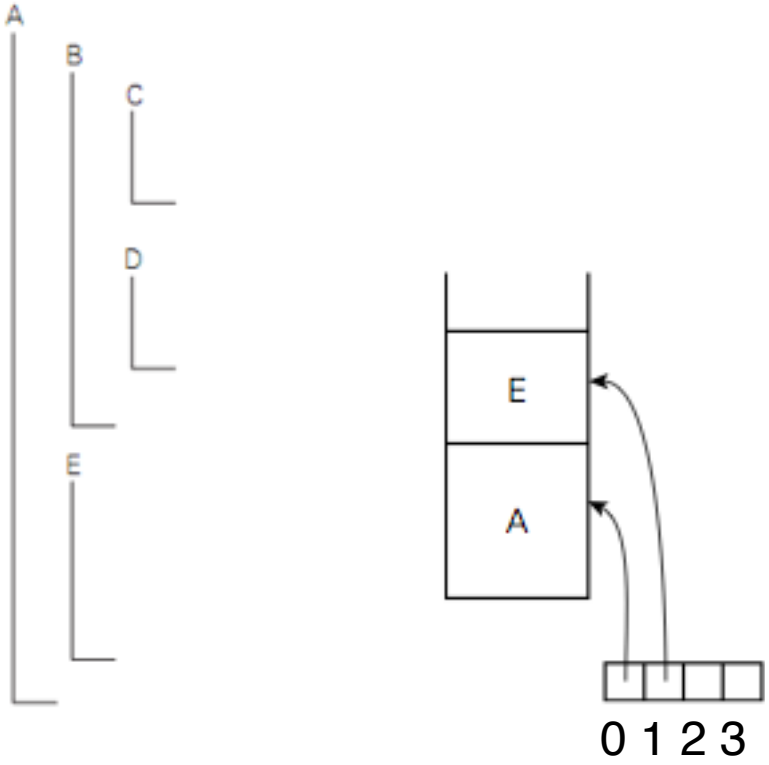
An array with one pointer for each nesting depth. Each display element records the most recently active AR for the corresponding depth

Cost:

In memory: reads AR address in one read

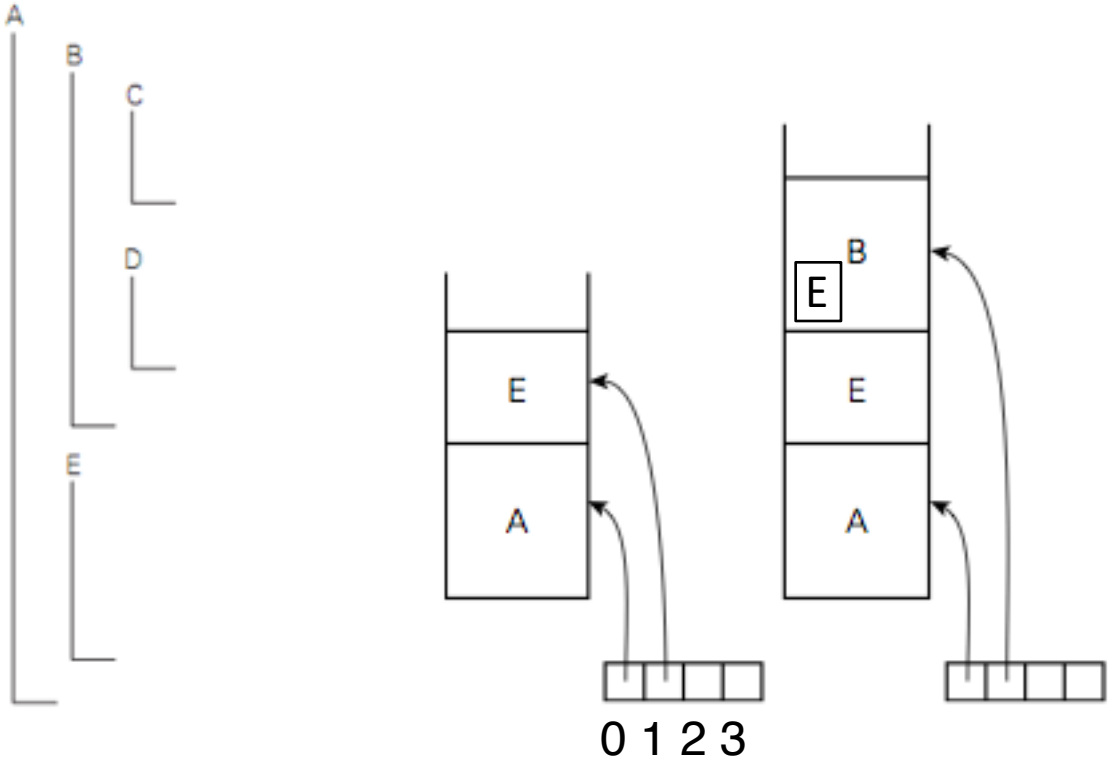
Display

Callee (at depth k) saves kth display element on stack, and replace that element with its AR



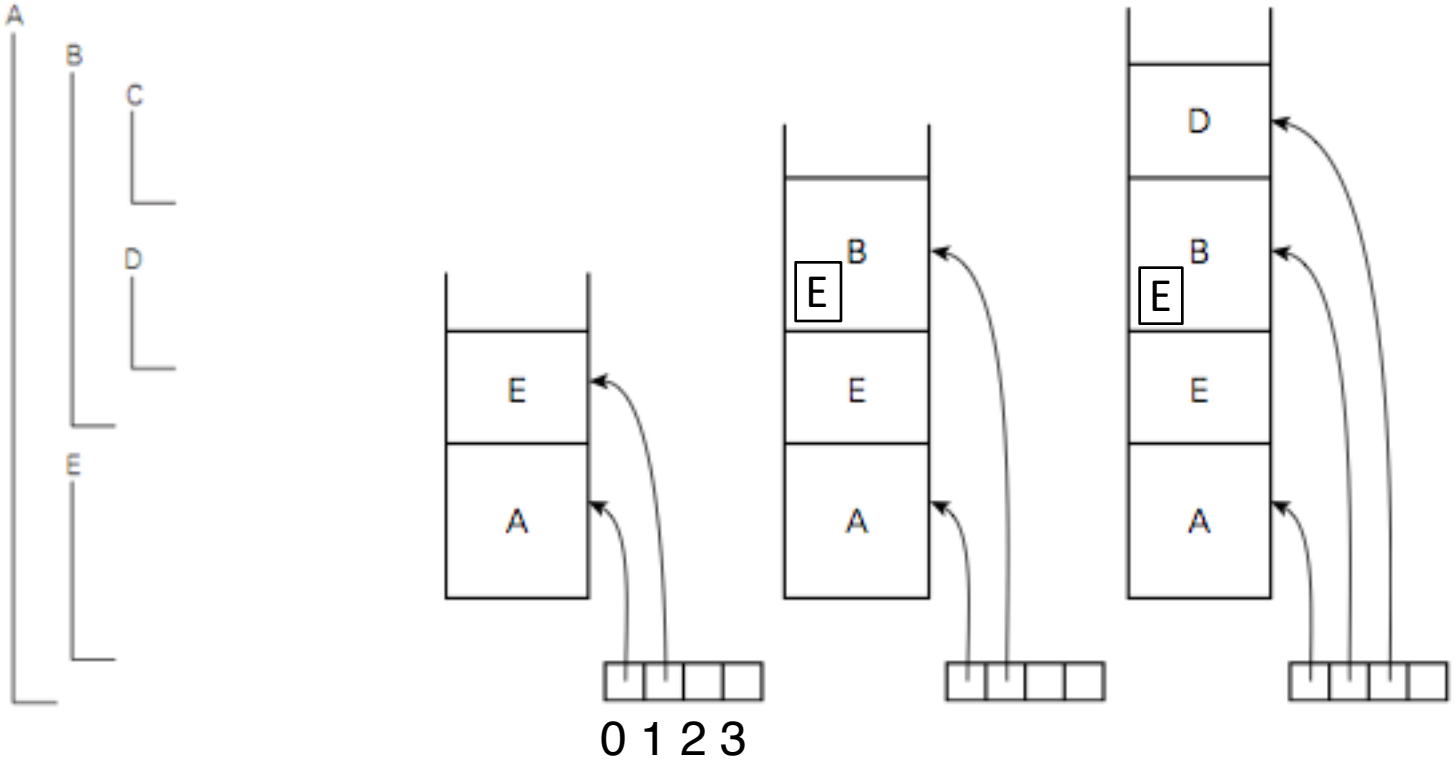
Display

Callee (at depth k) saves kth display element on stack, and replace that element with its AR



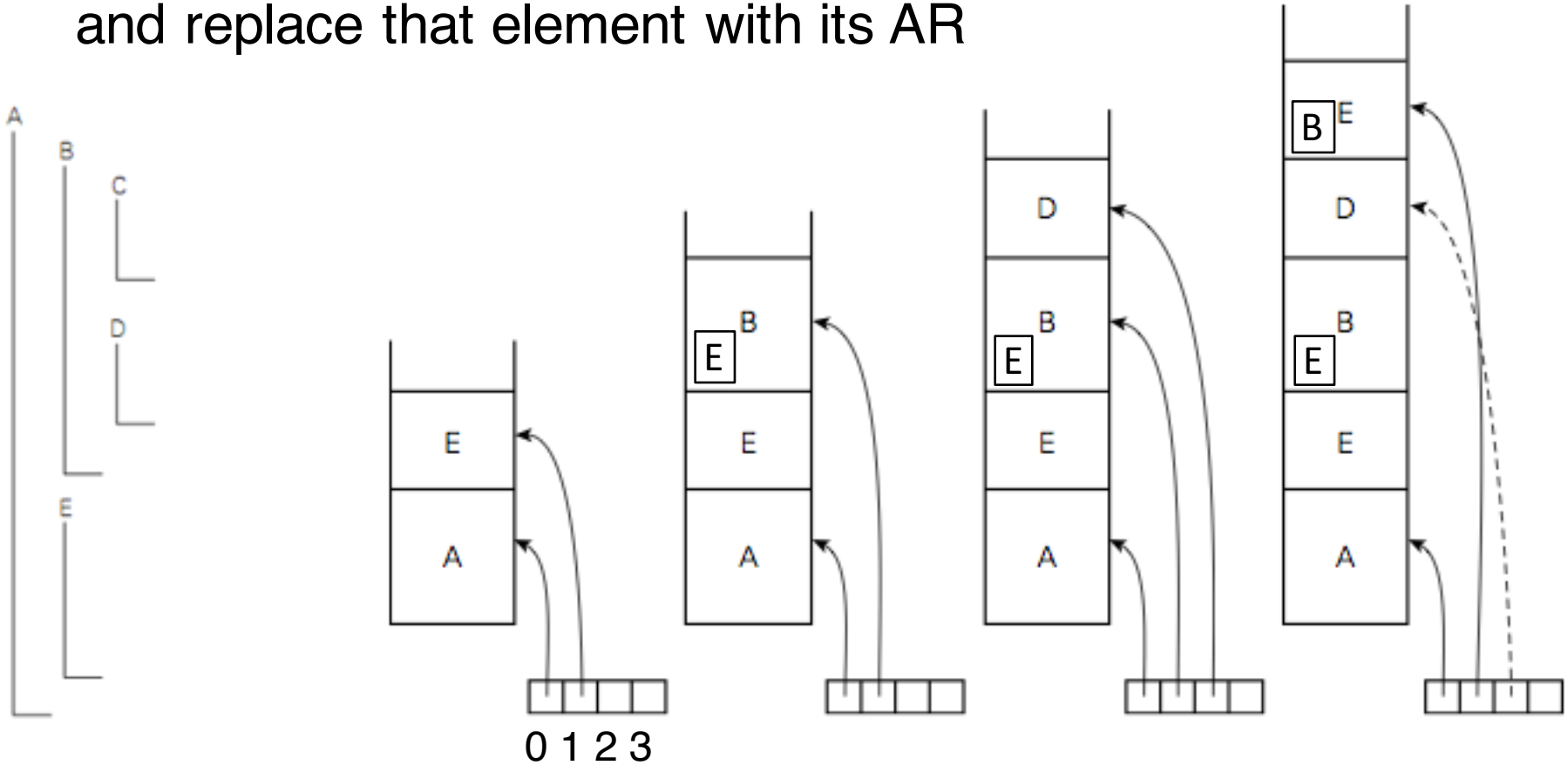
Display

Callee (at depth k) saves kth display element on stack, and replace that element with its AR



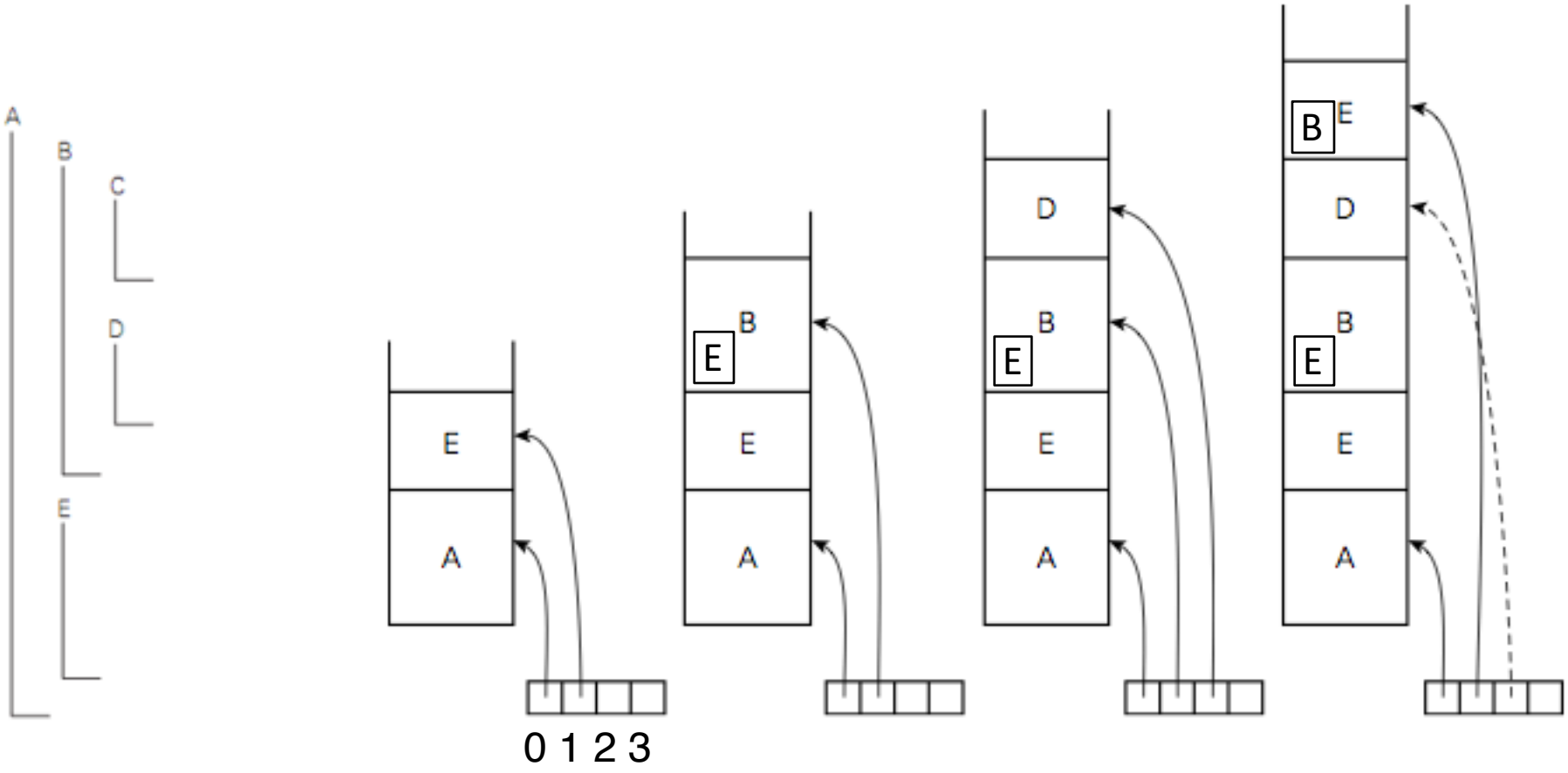
Display

Callee (at depth k) saves kth display element on stack, and replace that element with its AR



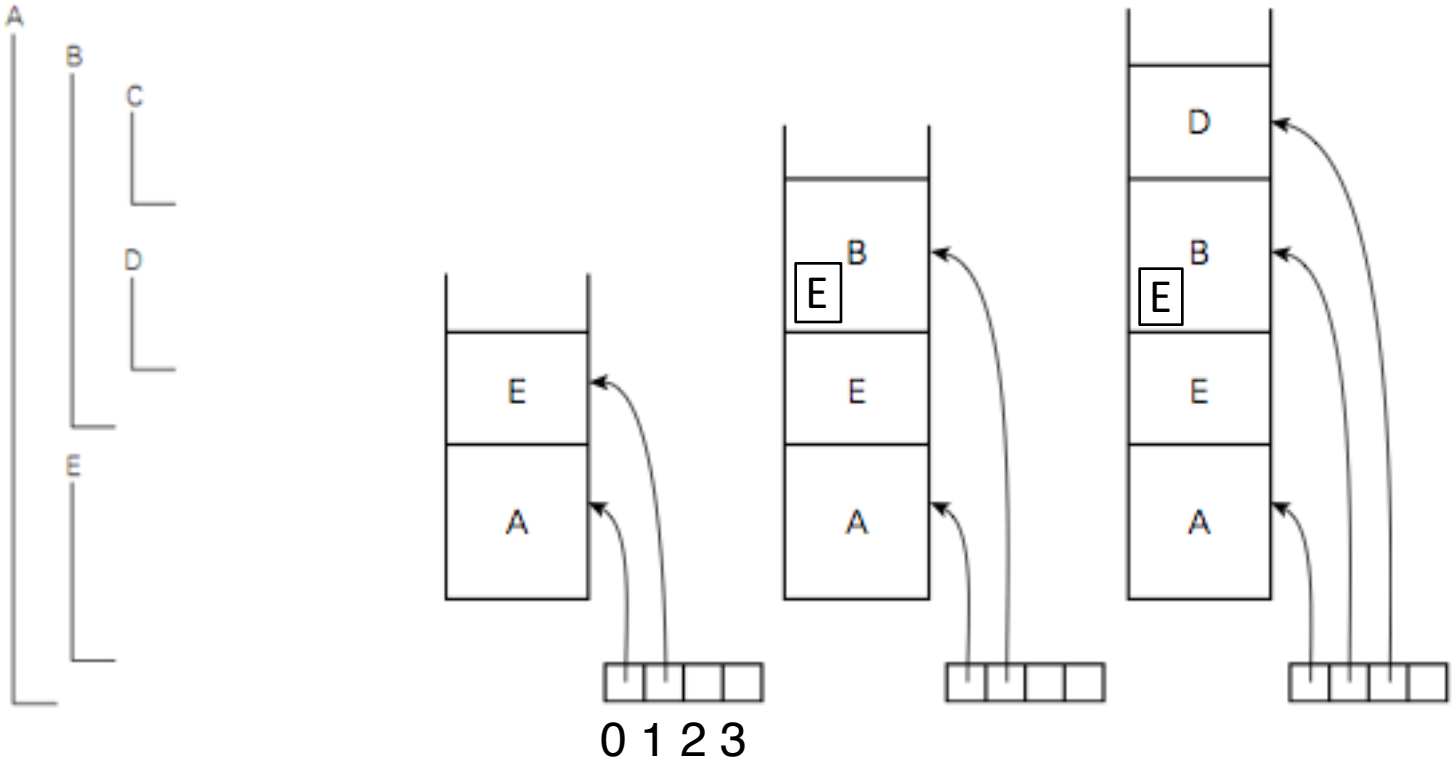
Display

When Callee returns, it restores the saved display element



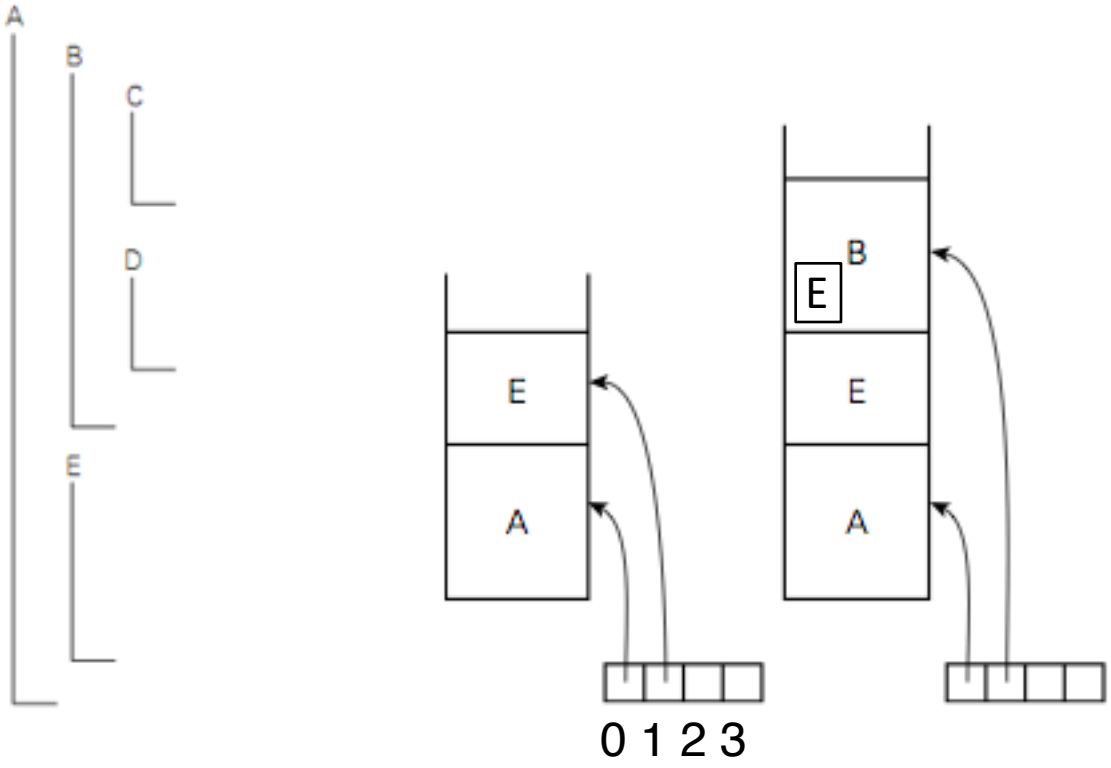
Display

When Callee returns, it restores the saved displav element



Display

When Callee returns, it restores the saved displav element



Display

When Callee returns, it restores the saved displav element