Subroutines

Parameters Passing

Methods of Passing Parameters

- In Registers
  Must document registers used
  Limited number of registers
  Uses fast on-chip registers
  Dictates register usage

- In a Parameter Block
  Parameters placed in a data structure
  Location of structure passed in a register
  Only uses one register
  Unlimited number of parameters
  Uses slow off-chip memory

More Parameter Passing Methods

- On the Stack
  Push parameters onto the stack
  Unlimited number of parameters
  Uses slow off-chip memory
  Don’t have to manage memory
  Need a calling convention

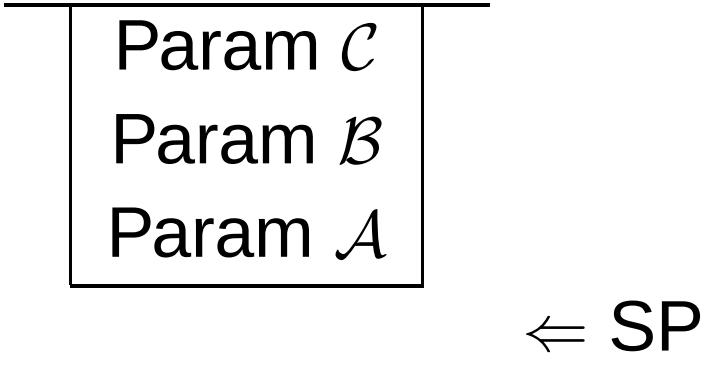
	C/C++	Pascal
order parameters pushed onto stack	reverse order	given order
who clears stack	caller	subroutine
return value	in a register	on the stack

The Stack Frame

- Normally used by Compilers
- Creates a hole in the stack for private use by the subroutine (local variables)
- Requires a *base* register to point to the start of the frame (SP points to the end)
- Subroutine parameters kept above the frame
- Local's stored inside the frame
- Stack can still be used for temporary storage

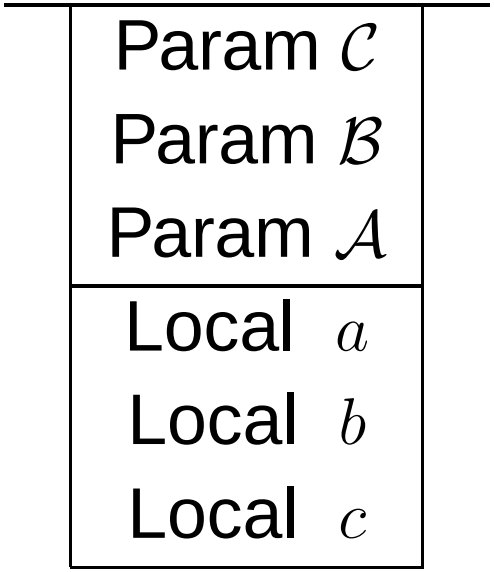
Operation of a Stack Frame

Operation of a Stack Frame



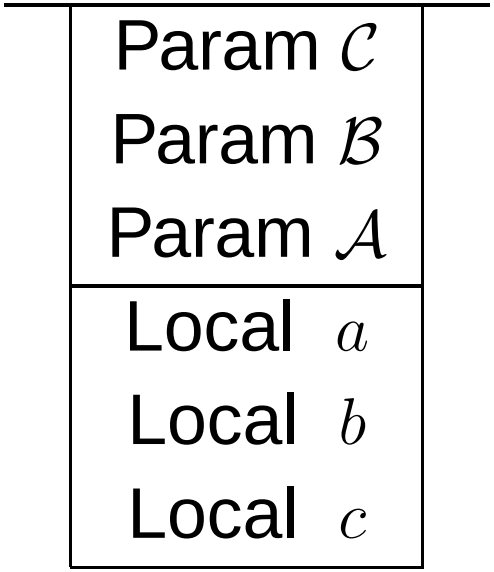
1 Push parameters ($\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$)

Operation of a Stack Frame



1. Push parameters ($\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$)
2. Reserve space for local variable (a, b, and c)
   Use R12 as Frame Pointer (FP)

Operation of a Stack Frame

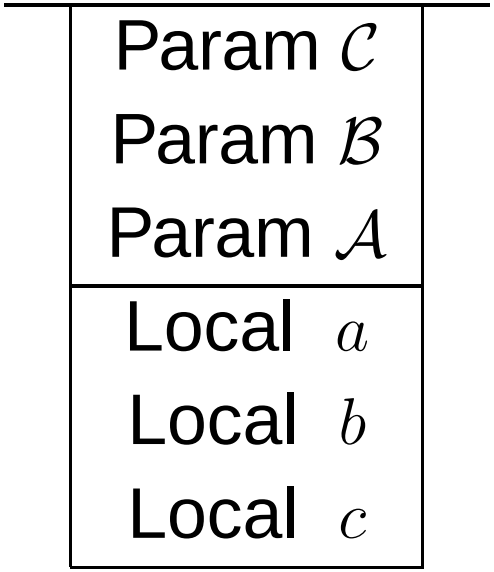


1. Push parameters ($\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$)
2. Reserve space for local variable (a, b, and c)

   Use R12 as Frame Pointer (FP)

   Stack can be used as normal

Operation of a Stack Frame



[R12, #8]

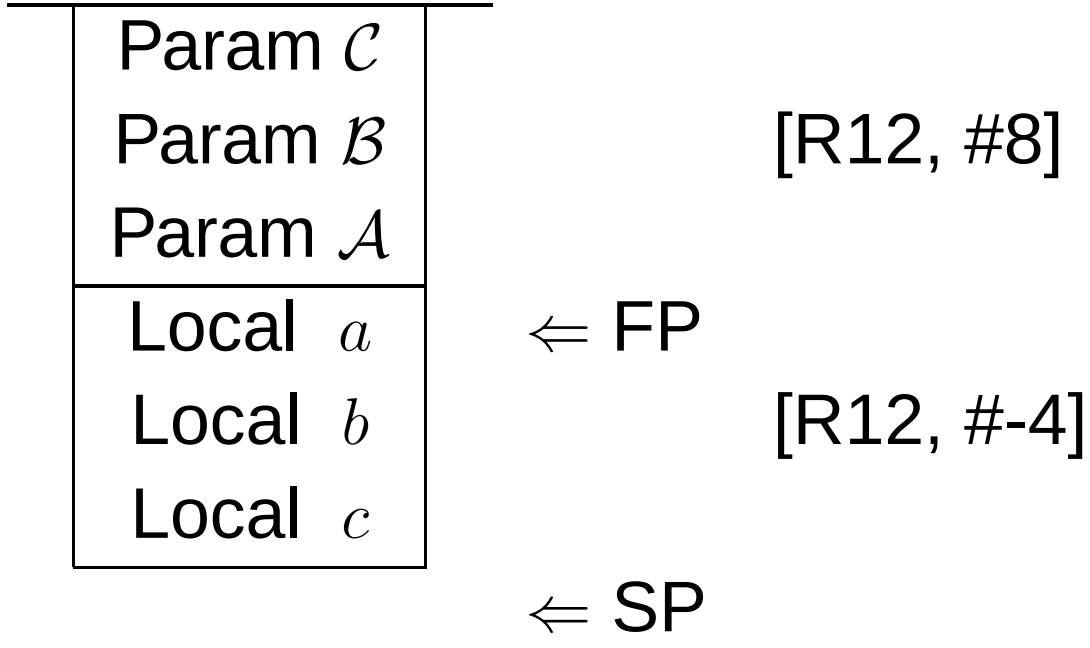
1. Push parameters ($\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$)
2. Reserve space for local variable (a, b, and c)

   Use R12 as Frame Pointer (FP)

   Stack can be used as normal

   Use positive offset to access parameters

Operation of a Stack Frame



1. Push parameters ($\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$)
2. Reserve space for local variable (a, b, and c)

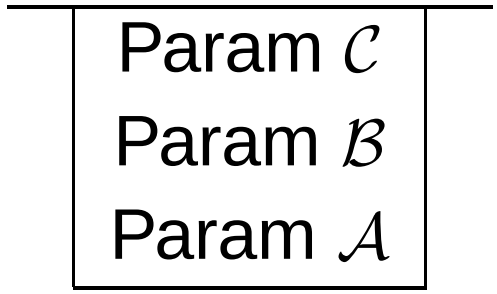
Use R12 as Frame Pointer (FP)

Stack can be used as normal

Use positive offset to access parameters

Use negative offset to access locals

Operation of a Stack Frame



[R12, #8]

1. Push parameters ($\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$)
2. Reserve space for local variable (a, b, and c)
   Use R12 as Frame Pointer (FP)
   Stack can be used as normal
3. Return from subroutine / Clear Locals

Operation of a Stack Frame

1. Push parameters ($\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$)
2. Reserve space for local variable (a, b, and c)
   Use R12 as Frame Pointer (FP)
   Stack can be used as normal
3. Return from subroutine / Clear Locals
4. Clear Stack / Pop parameters

Types of Parameter Passed

- Pass by value (most common)
  Value of the variable is used

- Pass by reference (common)
  Location (address) of variable is used
  Allows the subroutine to modify the original variable

- Pass by name (rare)
  A string holding the name of the data field is passed
  Subroutine uses string to access data (table or field)

Summary of Parameter Passing

- Methods of passing parameters
  Pass in Register
  Pass in Parameter Block
  Pass on Stack

- The Stack Frame
  Parameter Block on the Stack
  Used by Compilers

- Type of parameter passed
  Pass by value
  Pass by reference
  Pass by name