



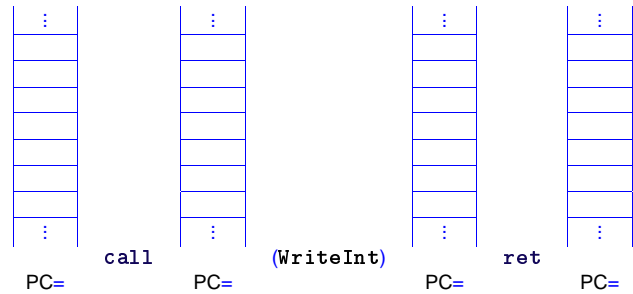
## Procedure Call – Example with Two Local Variables (2)

call-writeint.ass:

```

; WriteInt(n, 6); /*=printf("%6d", n);*/
load   n      ; /* Push(n) */ /* AC=Mem[n] */
incsp  #1     ; /* SP=SP+1 */
store  !0     ; /* Mem[SP]=AC */
load   #6     ; /* Push(#6) */ /* AC=6 */
incsp  #1     ; /* SP=SP+1 */
store  !0     ; /* Mem[SP]=AC */
call   WriteInt; /* SP=SP+1; Mem[SP]=PC;

```



## Non-void Function – Example with One Return Value (1)

```

RV     = -3   ; int Log10(
x      = -2   ; unsigned int x) {
;
;
Logx   = 0     ; unsigned int Logx;
NLocs  = 1     ;
;
Log10:
incsp  #NLocs ; /* SP=SP+1 */
load   !x     ; /* AC=Mem[SP-2] */
cmp    #0     ; /* compare AC,0 */
beq    Lendif ;
...
Lendif:
;
load   !Logx  ; return Logx; /* AC=Mem[SP] */
store  !RV    ; /* Mem[SP-3]=AC */
incsp  #-NLocs; } /*Log10()*/ /* SP=SP-1 */
ret    ; /* PC=Mem[SP]; SP=SP-1 */
...

```

Procedure Log10() is available in module InOut.ass

## Non-void Functions in PeANUt

- we implement local variables via the stack
  - good: economy, privacy, recursion and sharable, re-entrant code and multithread-safe
  - bad: we don't have stack-indexed (or stack-indirect) addressing modes in PeANUt
- inside procedures:
  - we have to increment the SP by number of local variables just after entry (first instruction within procedure)
  - and decrement SP by number of local variables just before (each) ret
  - parameters, local variables and return values (RVs) have stack offsets
- non-void function call is similar, but it must first make (empty) slot for the return value, which gets accessed after call before it is popped

## Non-void Function – Example with One Return Value (2)

call-log10.ass:

```

; log = Log10( 511 );
incsp  #1     ; /* SP=SP+1 */ /* make RV slot */
Push   (#511) ; /* AC=511; SP=SP+1; Mem[SP]=AC */
call   Log10  ; /* SP=SP+1; Mem[SP]=PC; PC=Log10 */
Pop    (1)    ; /* SP=SP-1 */
load   !0     ; /* AC=Mem[SP] */ /*store RV*/
store  log    ; /* Mem[Log]=AC */
incsp  #-1    ; /* SP=SP-1 */ /*pop RV slot*/

```

