The Calling Sequence The calling sequence now comprises approximately the following steps. When a procedure is called,

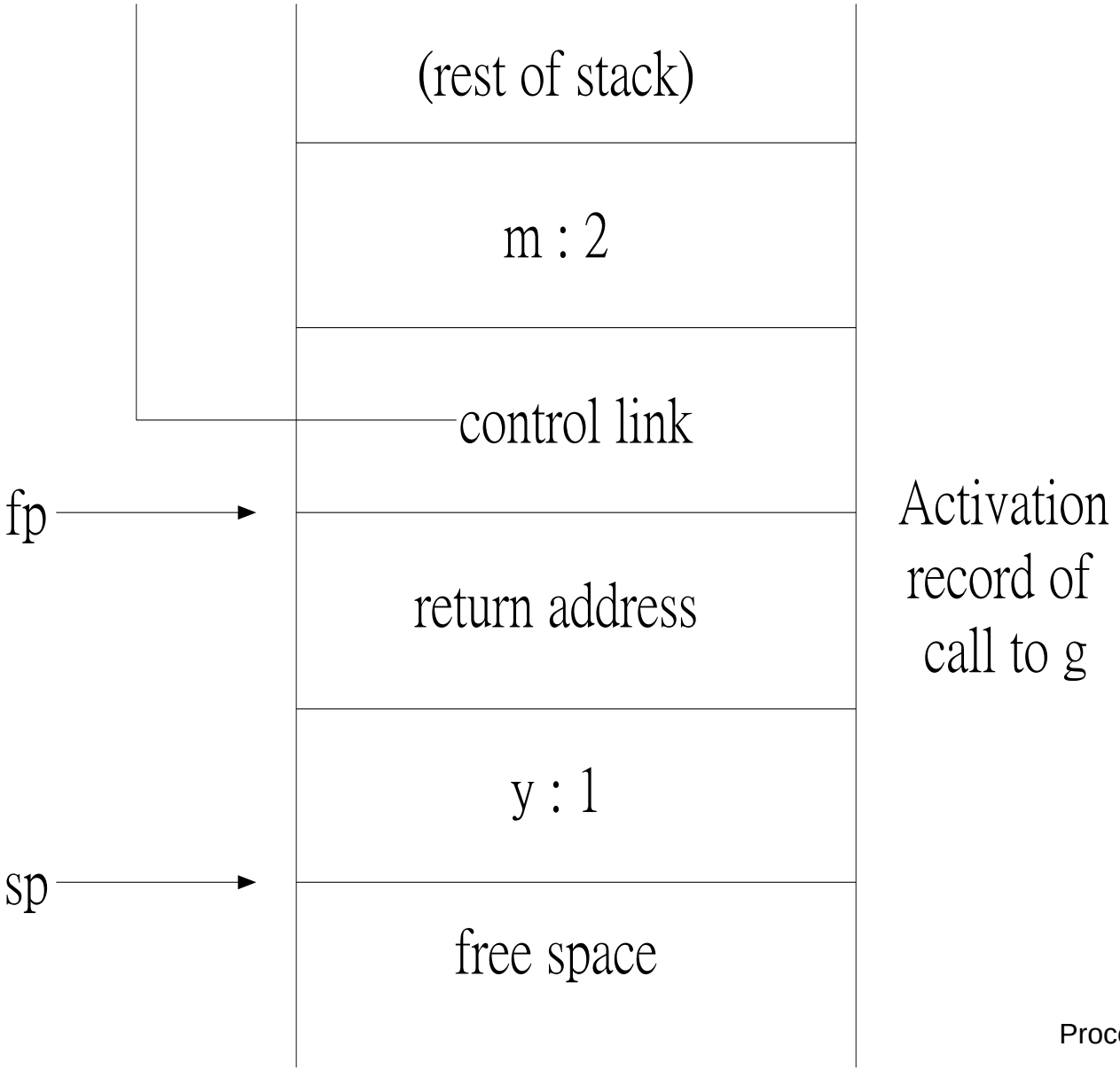
1. Compute the arguments and store them in their correct positions in the new activation record of the procedure (pushing them in order onto the runtime stack will achieve this).
2. Store (push) the fp as the control link in the new activation record.
3. Change the fp so that it points to the beginning of the new activation record (if there is an sp, copying the sp into the fp at this point will achieve this).
4. Store the return address in the new activation record (if necessary).
5. Perform a jump to the code of the procedure to be called.

<<Procedure Call.ptt>>

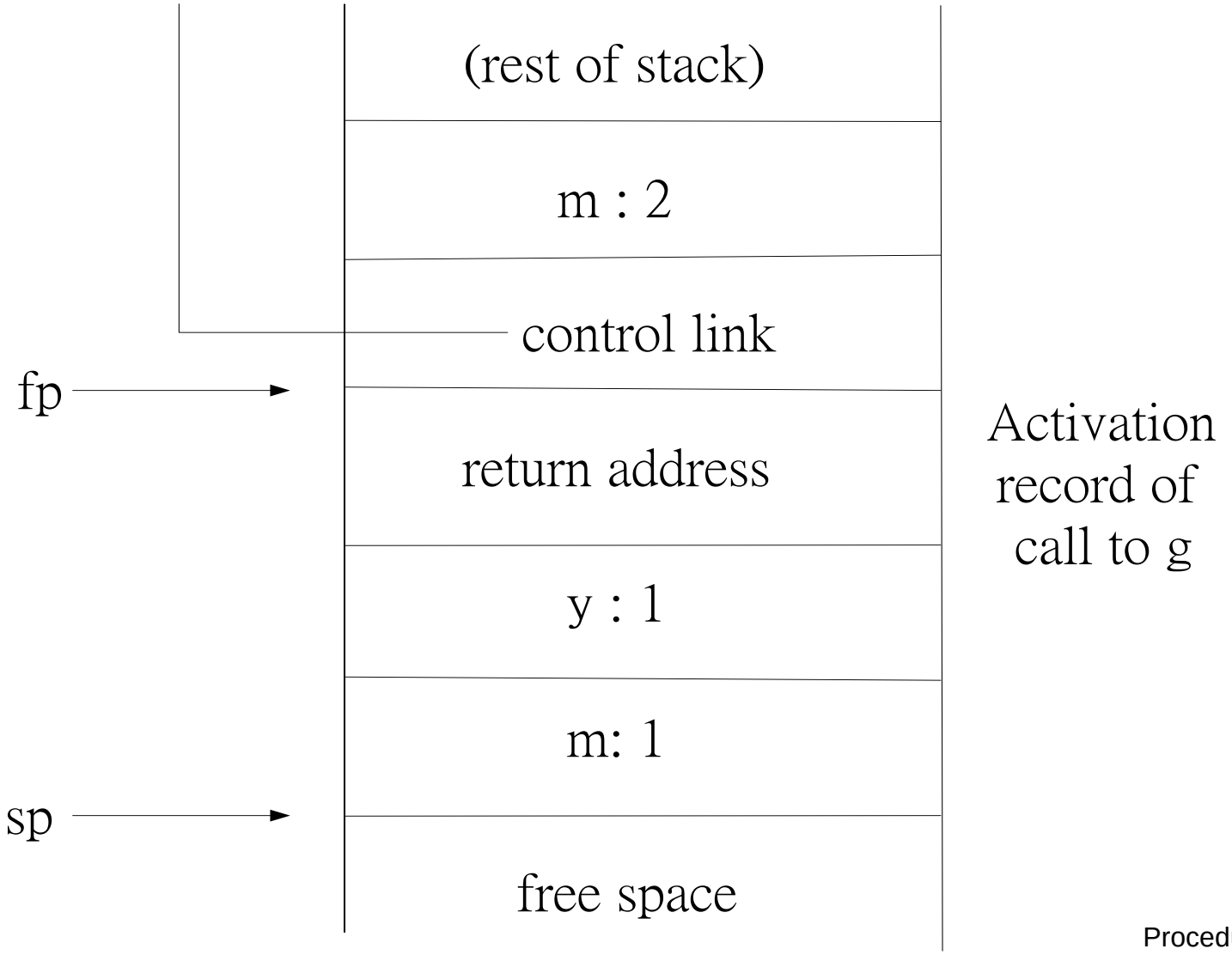
When a procedure exits,

1. Copy the fp to the sp.
2. Load the control link into the fp.
3. Perform a jump to the return address.
4. Change the sp to pop the arguments.

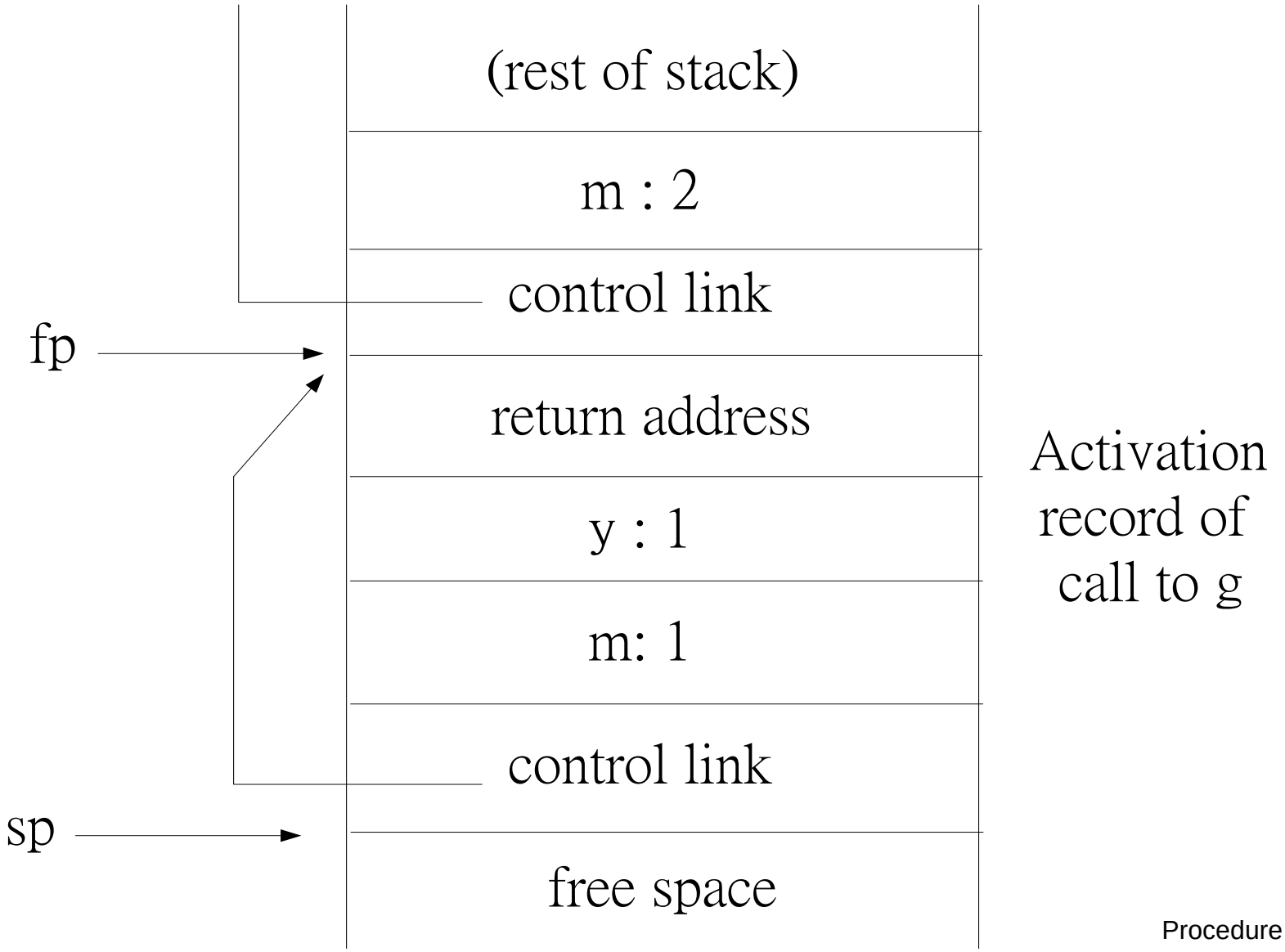
Consider the situation just before the last call to g in Figure 7.6(b):



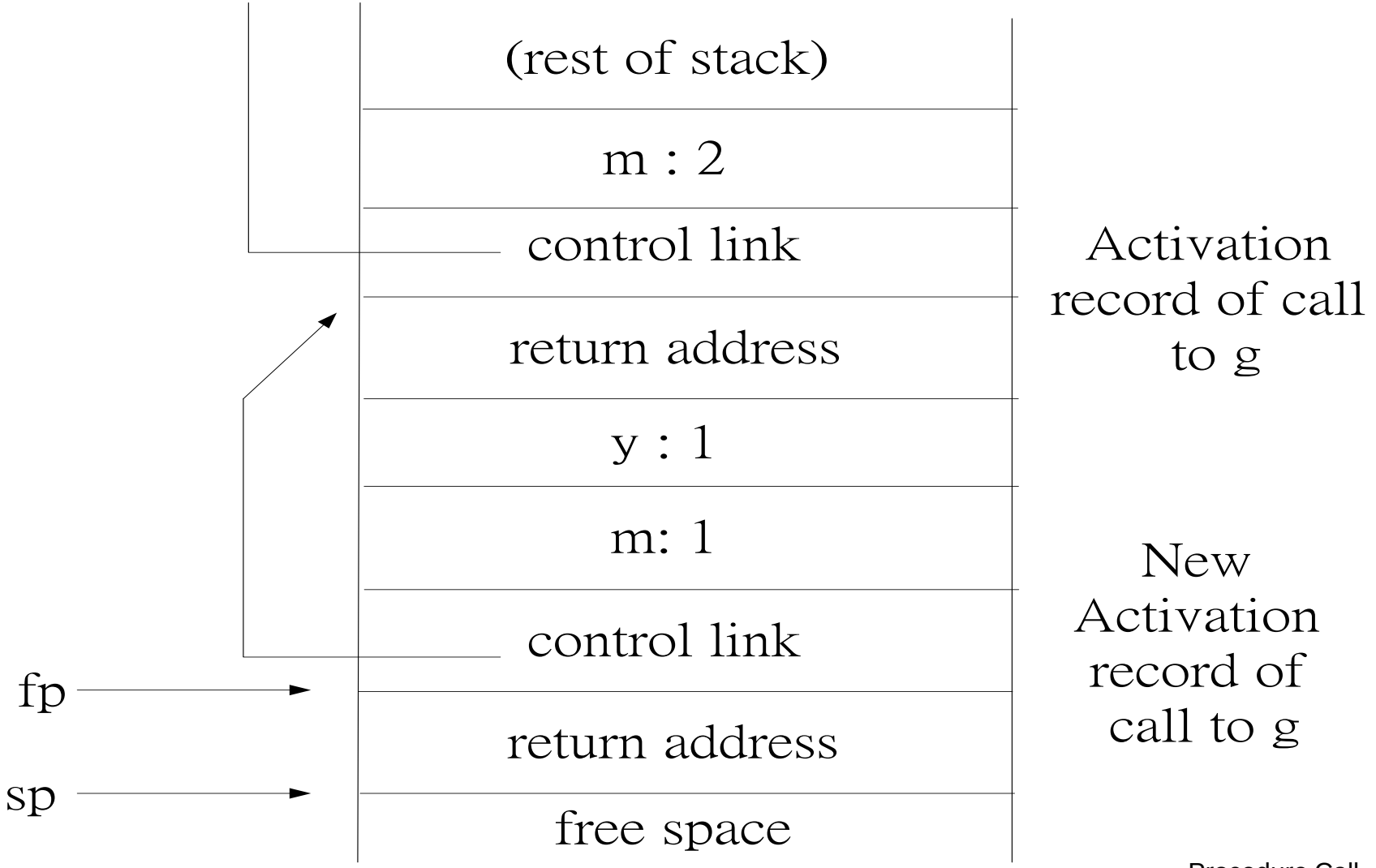
As the new call to g is made, first the value of parameter m is pushed onto the runtime stack:



then the fp is pushed onto the stack:



Now the sp is copied into the fp, the return address is pushed onto the stack, and the jump to the new call of g is made:



Finally, g allocates and initializes the new y on the stack to complete the construction of the new activation record: