WE12.1

Bookmark this page

Video explanation of solution is provided below the problem.

Procedures and Stacks

11/11 points (ungraded)

You've been commissioned by a government agency to reverse-engineer a mysterious procedure found on the disk of a Beta system used by a cyber-terrorist cell. You've given an incomplete copy of the C source language for the function **f** (shown below), as well as its complete translation to Beta assembly code:

```
// Mystery function:
int f(int x) {
  int a = x&5;  // bitwise AND

  if (x == 0) return 0;
  else return ?????;

}
```

```
f:   PUSH(LP)
     PUSH(BP)
     MOVE(SP, BP)
     ALLOCATE(1)
     PUSH(R1)

     LD(BP, -12, R0)
     ANDC(R0, 5, R1)
     ST(R1, 0, BP)

xx:  BEQ(R0, bye)

     SUBC(R0, 1, R0)
     PUSH(R0)
yy:  BR(f, LP)
     DEALLOCATE(1)

     LD(BP, 0, R1)
     ADD(R1, R0, R0)

bye: POP(R1)
     MOVE(BP,SP)
     POP(BP)
     POP(LP)
     JMP(LP)
```

1. What is the missing expression shown in the C code as **"?????"**

   - ( ) f(x)
   - ( ) f(x-1)
   - ( ) a + f(x+1)
   - ( ) a + f(x)
   - (•) a + f(x-1) ✔

2. Is the value of the local variable **a** stored in the stack frame of the Beta program? If so, give its offset relative to the contents of **BP**; otherwise, write "None":

   Offset of a, or None: 0 ✔

3. Give the 32-bit binary translation of the BR instruction tagged **yy**

   opcode (6 bits): 0b 011100 ✔

   Rc (5 bits): 0b 11100 ✔

   Ra (5 bits): 0b 11111 ✔

   literal (16 bits): 0b 1111111111110000 ✔

The function **f** is called from an external main program, and the machine is halted when a recursive call to **f** is about to execute the **BEQ** instruction tagged **xx**. The BP register of the halted machine contains **0×174**, and the hex contents of a region of memory location are shown below.

13C:	\(7\)
140:	\(7\)
144:	\(5C\)
148:	\(D4\)
14C:	\(5\)
150:	\(3\)
154:	\(6\)

Calculator

158:	\(A4\)
15C	\(14C\)
160	\(4\)
164	\(5\)
168	\(5\)
16C	\(A4\)
170	\(160\)
\(BP\rightarrow\)174:	\(5\)
178	\(4\)

4. What is the value in **SP**?

   HEX contents of SP: 0x 17C ✔

5. What is the value stored in the local variable **a** in the current stack frame?

   HEX value of a: 0x 5 ✔

6. What is the address of the **BR** instruction that made the original call to **f** from the external main program?

   Address of BR for original call: 0x 58 ✔

7. What value is currently in the PC?

   HEX contents of PC: 0x 90 ✔

   The summer intern working for you, after looking at the assembly code for f, argues that the cyber-terrorist group that wrote this code isn't very clever. He argues that one could simply delete four instructions from the assembly-language program -- a **LD**, a **ST**, an **ALLOCATE**, and a **MOVE** -- and the program would continue to work as before (but faster and using less space).

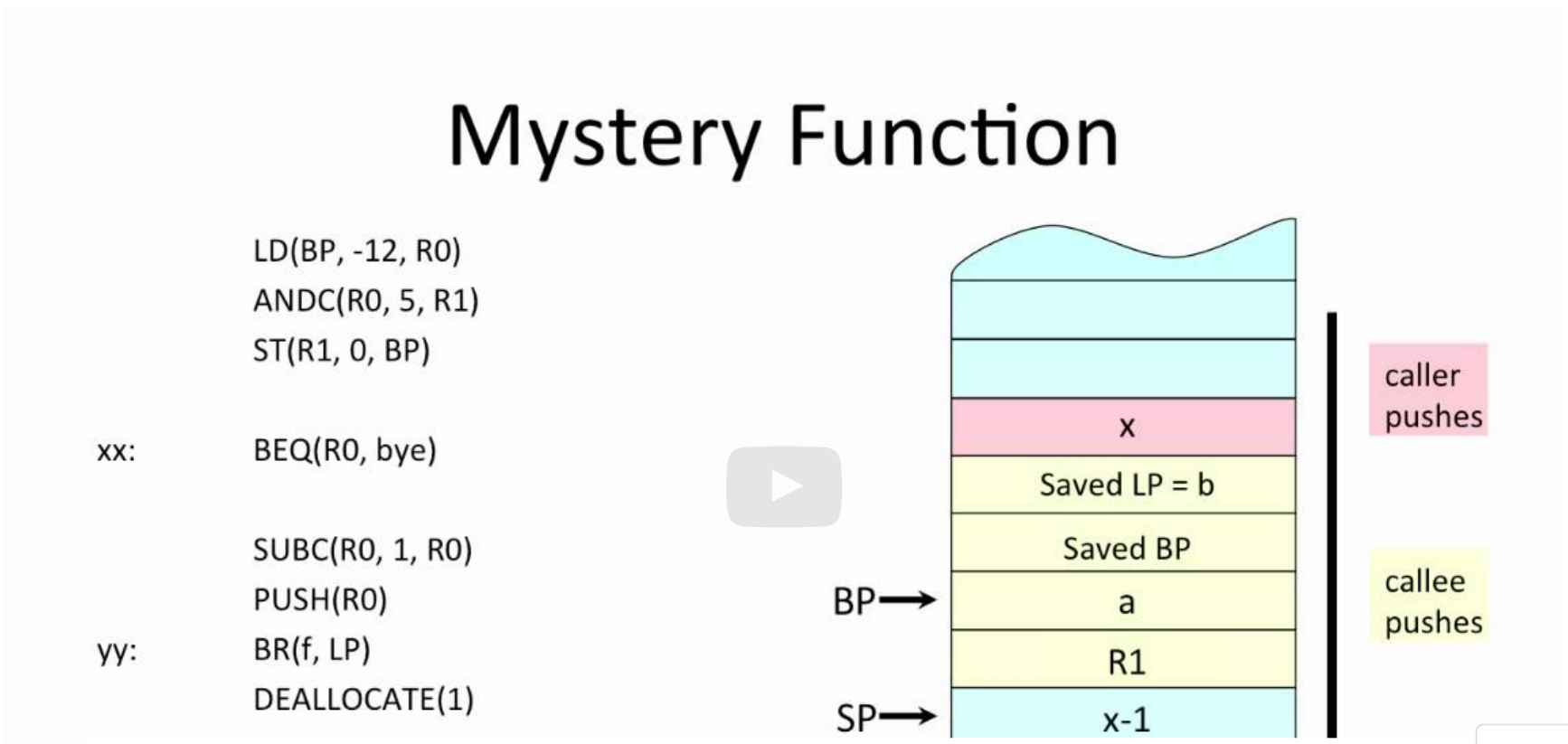
8. Is he right? Can one in fact delete four lines of code as described and still have a working program?
   Can one optimize by deleting 4 such lines?

   - (●) YES
   - ( ) NO

   ✔

Submit

Procedures and Stacks



(Caption will be displayed when you start playing the video.)

Calculator

Video
Download video file

Transcripts
Download SubRip (.srt) file
Download Text (.txt) file

Discussion

Topic: 12. Procedures and Stacks / WE12.1

Hide Discussion

Add a Post

Show all posts by recent activity

Stack record in Bsim
I wonder if we could "simulate" the stack record and see it in Bsim. If I copy the assembly code in this example and paste it in Bsim, a... 3

[STAFF] 16 bit answer doesn't quite fit
Hi Silvina, Could you please widen the area for the answer to the 16 bit literal in WE12.1(C) and LE12.3(A) just a little bit? Enjoying my ... 1

Previous Next

Calculator