**rbpf** (https://github.com/qmonnet/rbpf) machine can’t load code which uses global variables because it doesn't do relocations.

**The task:**

You need to modify the rbpf code to be able to execute the following eBPF program:

|  |
| --- |
| // bpf\_test.c int g = 1;  int bpf\_prog(int \*b) {  int sum = 0;  sum += \*b + g;  g = sum;  return g;  } |

Compile with the following commands:

|  |
| --- |
| $ clang -target bpf -S -emit-llvm -c bpf\_test.c -o bpf\_test.ll  $ llc -march=bpf -filetype=obj bpf\_test.ll -o **bpf\_test.o** |

*It’s important that the code should be compiled without optimizations.*

**bpf\_test.o** **should be executed in rbpf machine.**

Below is a disassembled bpf\_test.o for example:

|  |
| --- |
| $ bpf-objdump -d bpf\_test.o    bpf\_test.o: file format elf64-bpfle      Disassembly of section .text:    0000000000000000 <bpf\_prog>:  0: 7b 1a f8 ff 00 00 00 00 stxdw [%fp+-8],%r1  8: b7 01 00 00 00 00 00 00 mov %r1,0  10: 63 1a f4 ff 00 00 00 00 stxw [%fp+-12],%r1  18: 79 a1 f8 ff 00 00 00 00 ldxdw %r1,[%fp+-8]  20: 61 11 00 00 00 00 00 00 ldxw %r1,[%r1+0]  28: 18 02 00 00 00 00 00 00 lddw %r2,0  30: 00 00 00 00 00 00 00 00  38: 61 23 00 00 00 00 00 00 ldxw %r3,[%r2+0]  40: 0f 31 00 00 00 00 00 00 add %r1,%r3  48: 61 a3 f4 ff 00 00 00 00 ldxw %r3,[%fp+-12]  50: 0f 13 00 00 00 00 00 00 add %r3,%r1  58: 63 3a f4 ff 00 00 00 00 stxw [%fp+-12],%r3  60: 61 a1 f4 ff 00 00 00 00 ldxw %r1,[%fp+-12]  68: 63 12 00 00 00 00 00 00 stxw [%r2+0],%r1  70: 61 20 00 00 00 00 00 00 ldxw %r0,[%r2+0]  78: 95 00 00 00 00 00 00 00 exit |