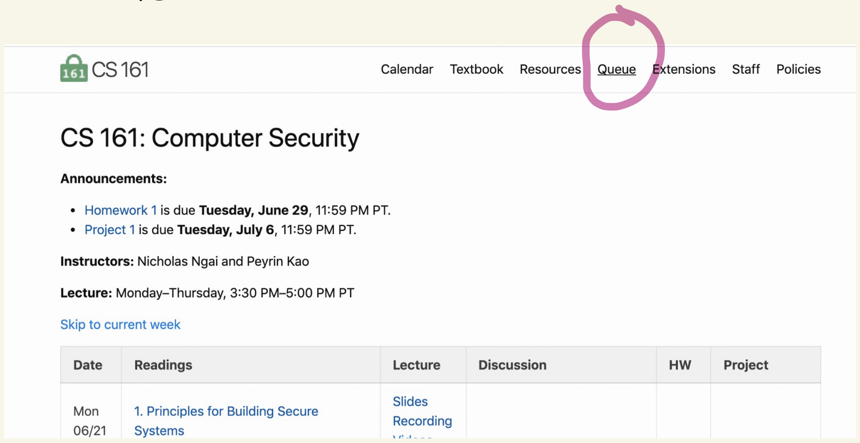
Logistics

- introduction
- office hours :



161 CS 161 Calendar Textbook Resources Queue Extensions Staff Policies

CS 161: Computer Security

Announcements:

- Homework 1 is due **Tuesday, June 29**, 11:59 PM PT.
- Project 1 is due **Tuesday, July 6**, 11:59 PM PT.

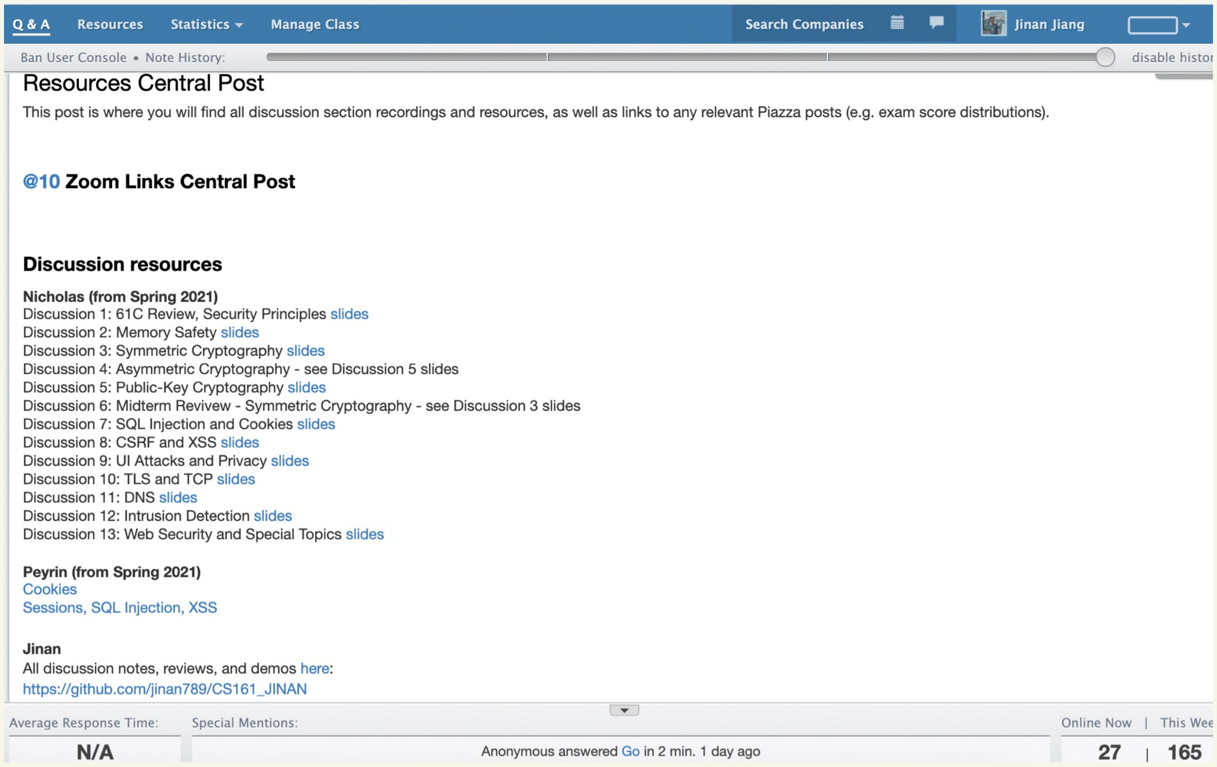
Instructors: Nicholas Ngai and Peyrin Kao

Lecture: Monday–Thursday, 3:30 PM–5:00 PM PT

Skip to current week

Date	Readings	Lecture	Discussion	HW	Project
Mon 06/21	1. Principles for Building Secure Systems	Slides Recording			

- project 1 released. Start early
- HW 1 released. use OH & Piazza for help
- All discussion material & demo :



Resources Central Post

This post is where you will find all discussion section recordings and resources, as well as links to any relevant Piazza posts (e.g. exam score distributions).

@10 Zoom Links Central Post

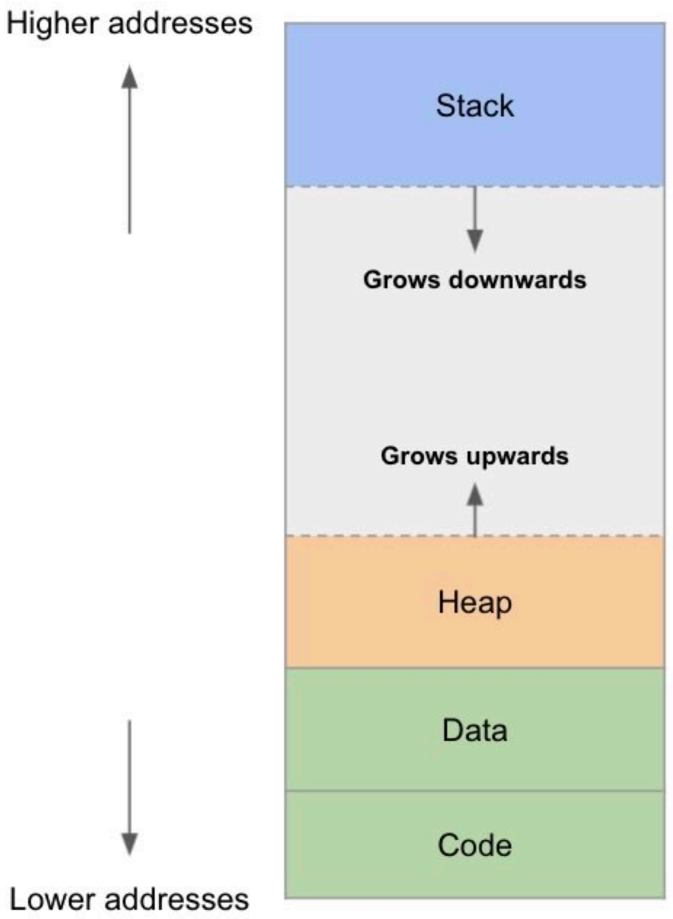
Discussion resources

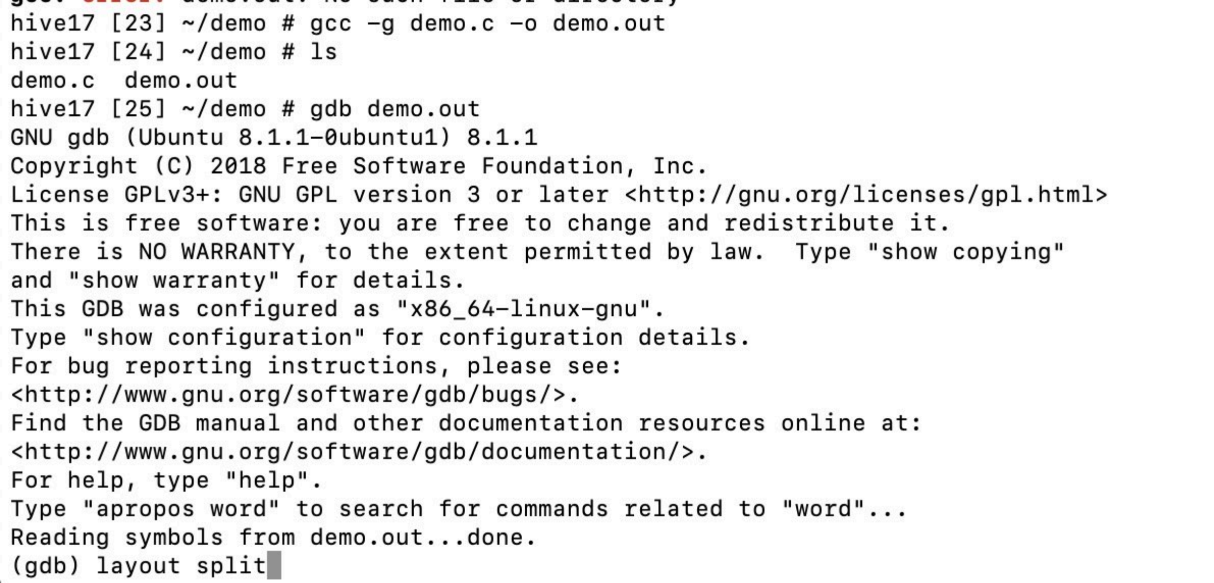
Nicholas (from Spring 2021)
Discussion 1: 61C Review, Security Principles slides
Discussion 2: Memory Safety slides
Discussion 3: Symmetric Cryptography slides
Discussion 4: Asymmetric Cryptography - see Discussion 5 slides
Discussion 5: Public-Key Cryptography slides
Discussion 6: Midterm Revivew - Symmetric Cryptography - see Discussion 3 slides
Discussion 7: SQL Injection and Cookies slides
Discussion 8: CSRF and XSS slides
Discussion 9: UI Attacks and Privacy slides
Discussion 10: TLS and TCP slides
Discussion 11: DNS slides
Discussion 12: Intrusion Detection slides
Discussion 13: Web Security and Special Topics slides

Peyrin (from Spring 2021)
Cookies
Sessions, SQL Injection, XSS

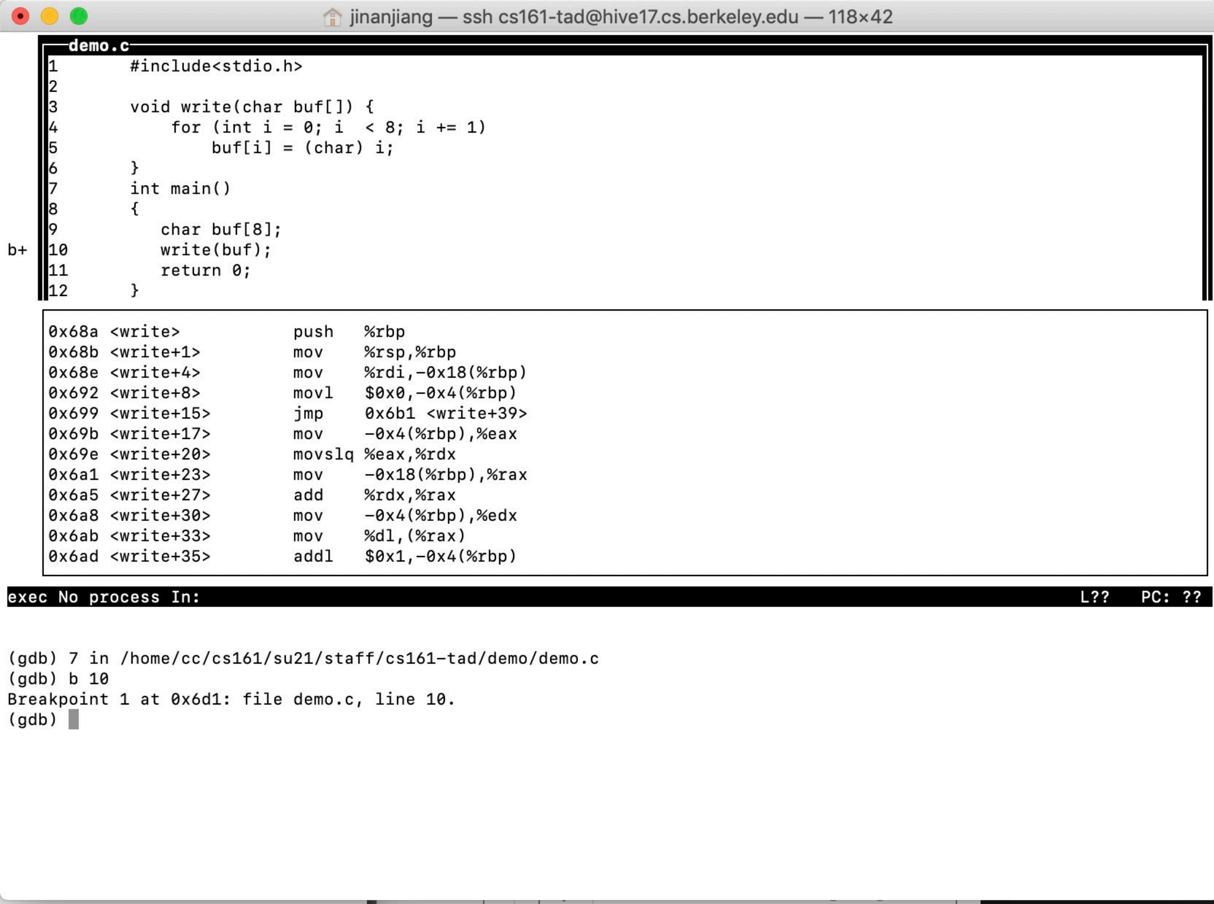
Jinan
All discussion notes, reviews, and demos here:
https://github.com/jinan789/CS161_JINAN

Average Response Time: N/A
Special Mentions: Anonymous answered Go in 2 min. 1 day ago
Online Now | This Wee
27 | 165


Higher addresses
Stack
Grows downwards
Grows upwards
Heap
Data
Code
Lower addresses



```
gcc: error: demo.out: No such file or directory
hive17 [23] ~/demo # gcc -g demo.c -o demo.out
hive17 [24] ~/demo # ls
demo.c  demo.out
hive17 [25] ~/demo # gdb demo.out
GNU gdb (Ubuntu 8.1.1-0ubuntu1) 8.1.1
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from demo.out...done.
(gdb) layout split
```



jinanjiang — ssh cs161-tad@hive17.cs.berkeley.edu — 118×42

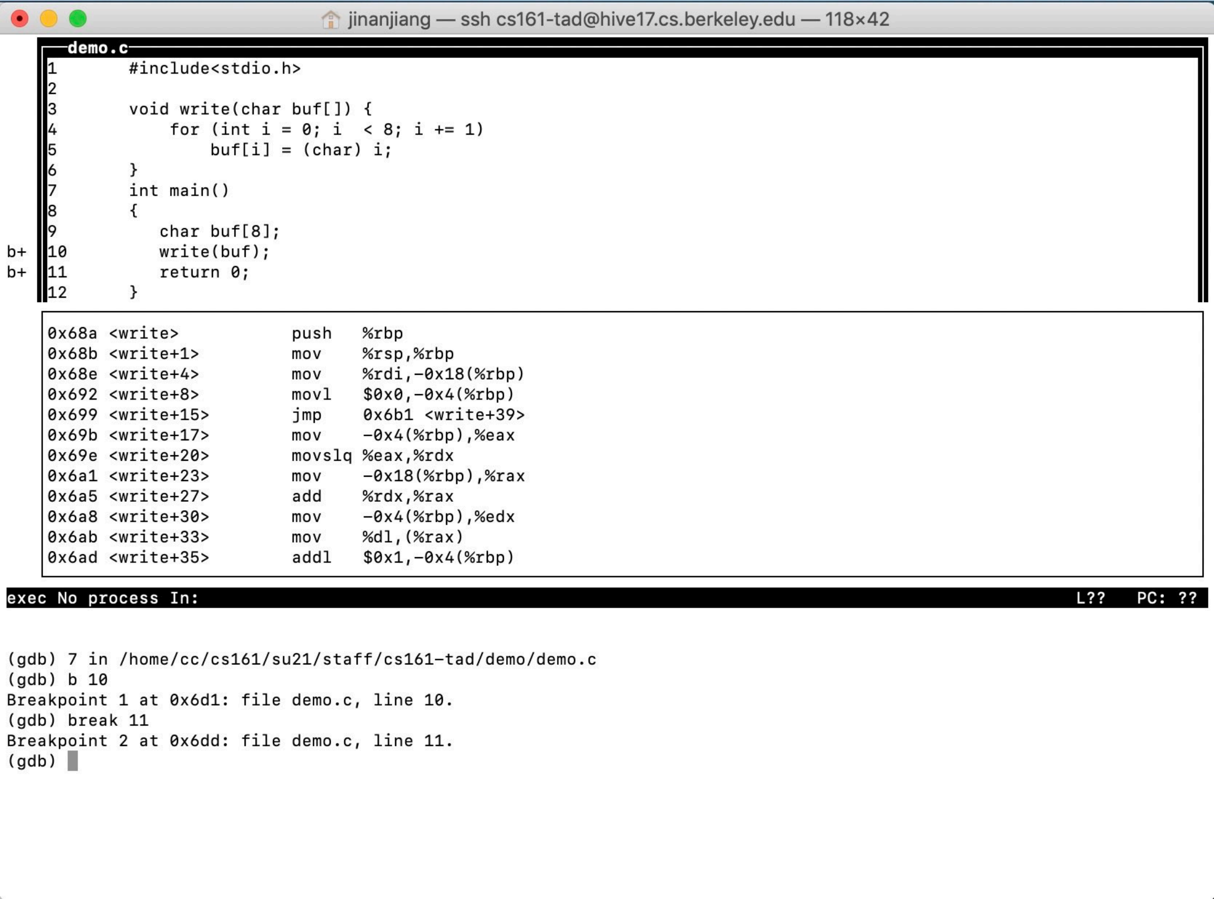
```
demo.c
        #include<stdio.h>

        void write(char buf[]) {
            for (int i = 0; i  < 8; i += 1)
                buf[i] = (char) i;
        }
        int main()
        {
           char buf[8];
b+         write(buf);
           return 0;
        }
```

```
0x68a <write>            push   %rbp
0x68b <write+1>          mov    %rsp,%rbp
0x68e <write+4>          mov    %rdi,-0x18(%rbp)
0x692 <write+8>          movl   $0x0,-0x4(%rbp)
0x699 <write+15>         jmp    0x6b1 <write+39>
0x69b <write+17>         mov    -0x4(%rbp),%eax
0x69e <write+20>         movslq %eax,%rdx
0x6a1 <write+23>         mov    -0x18(%rbp),%rax
0x6a5 <write+27>         add    %rdx,%rax
0x6a8 <write+30>         mov    -0x4(%rbp),%edx
0x6ab <write+33>         mov    %dl,(%rax)
0x6ad <write+35>         addl   $0x1,-0x4(%rbp)
```

```
exec No process In:                                                     L??   PC: ??


(gdb) 7 in /home/cc/cs161/su21/staff/cs161-tad/demo/demo.c
(gdb) b 10
Breakpoint 1 at 0x6d1: file demo.c, line 10.
(gdb)
```



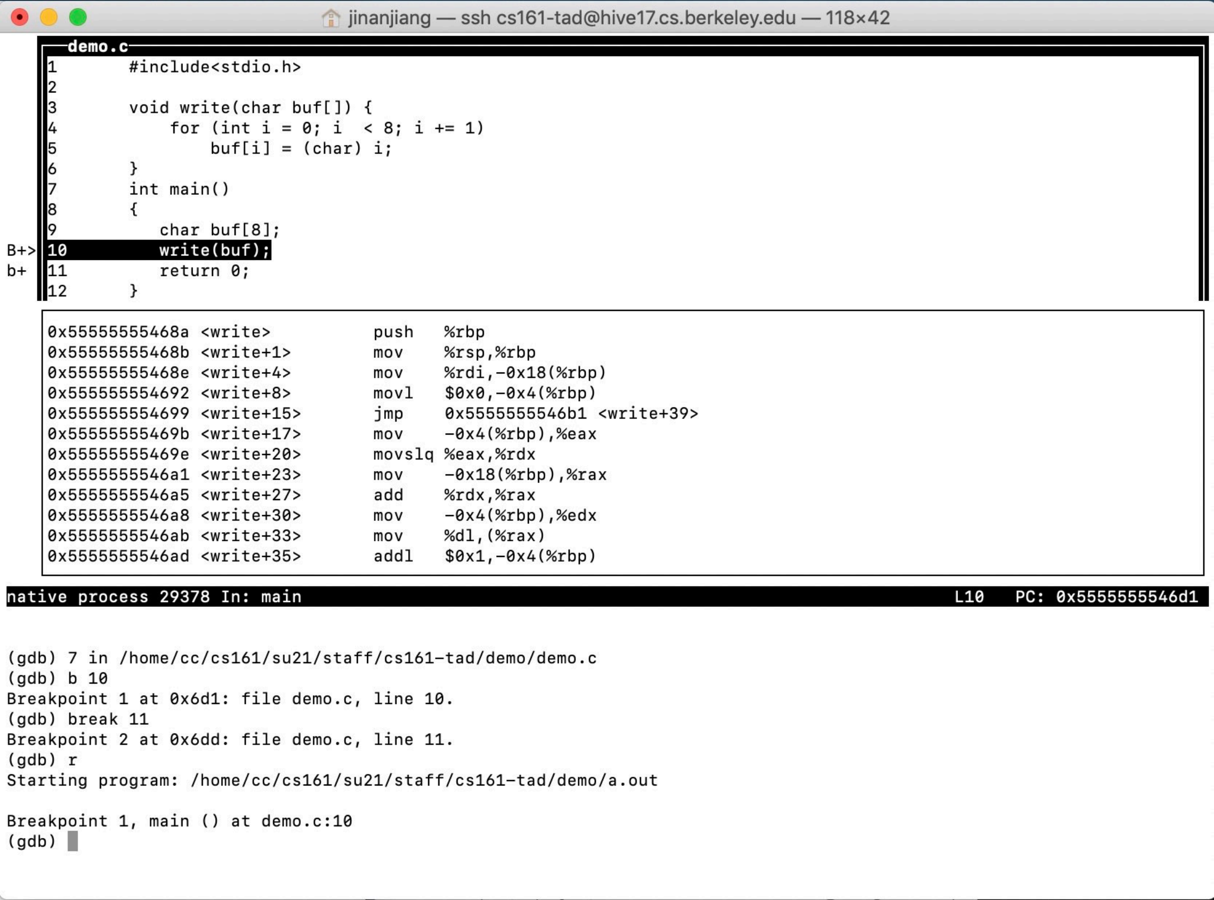
jinanjiang — ssh cs161-tad@hive17.cs.berkeley.edu — 118×42

```
demo.c
   1       #include<stdio.h>
   2
   3       void write(char buf[]) {
   4           for (int i = 0; i  < 8; i += 1)
   5               buf[i] = (char) i;
   6       }
   7       int main()
   8       {
   9          char buf[8];
b+ 10         write(buf);
b+ 11         return 0;
   12      }
```

```
0x68a <write>           push   %rbp
0x68b <write+1>         mov    %rsp,%rbp
0x68e <write+4>         mov    %rdi,-0x18(%rbp)
0x692 <write+8>         movl   $0x0,-0x4(%rbp)
0x699 <write+15>        jmp    0x6b1 <write+39>
0x69b <write+17>        mov    -0x4(%rbp),%eax
0x69e <write+20>        movslq %eax,%rdx
0x6a1 <write+23>        mov    -0x18(%rbp),%rax
0x6a5 <write+27>        add    %rdx,%rax
0x6a8 <write+30>        mov    -0x4(%rbp),%edx
0x6ab <write+33>        mov    %dl,(%rax)
0x6ad <write+35>        addl   $0x1,-0x4(%rbp)
```

```
exec No process In:                                                                          L??   PC: ??


(gdb) 7 in /home/cc/cs161/su21/staff/cs161-tad/demo/demo.c
(gdb) b 10
Breakpoint 1 at 0x6d1: file demo.c, line 10.
(gdb) break 11
Breakpoint 2 at 0x6dd: file demo.c, line 11.
(gdb) 
```



jinanjiang — ssh cs161-tad@hive17.cs.berkeley.edu — 118×42

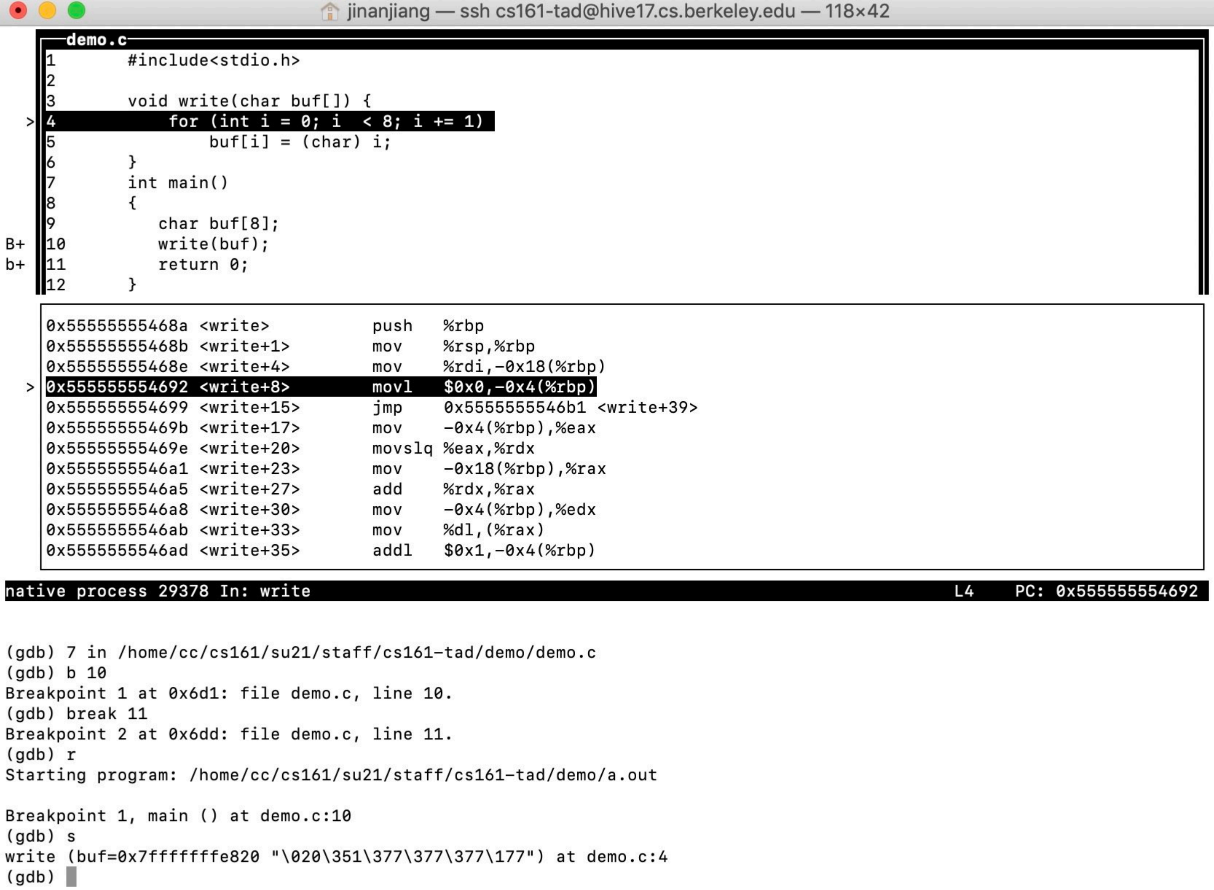
```
   demo.c
   1        #include<stdio.h>
   2
   3        void write(char buf[]) {
   4            for (int i = 0; i  < 8; i += 1)
   5                buf[i] = (char) i;
   6        }
   7        int main()
   8        {
   9           char buf[8];
B+>10           write(buf);
b+ 11           return 0;
   12       }
```

```
0x55555555468a <write>          push   %rbp
0x55555555468b <write+1>        mov    %rsp,%rbp
0x55555555468e <write+4>        mov    %rdi,-0x18(%rbp)
0x555555554692 <write+8>        movl   $0x0,-0x4(%rbp)
0x555555554699 <write+15>       jmp    0x5555555546b1 <write+39>
0x55555555469b <write+17>       mov    -0x4(%rbp),%eax
0x55555555469e <write+20>       movslq %eax,%rdx
0x5555555546a1 <write+23>       mov    -0x18(%rbp),%rax
0x5555555546a5 <write+27>       add    %rdx,%rax
0x5555555546a8 <write+30>       mov    -0x4(%rbp),%edx
0x5555555546ab <write+33>       mov    %dl,(%rax)
0x5555555546ad <write+35>       addl   $0x1,-0x4(%rbp)
```

```
native process 29378 In: main                                                        L10    PC: 0x5555555546d1


(gdb) 7 in /home/cc/cs161/su21/staff/cs161-tad/demo/demo.c
(gdb) b 10
Breakpoint 1 at 0x6d1: file demo.c, line 10.
(gdb) break 11
Breakpoint 2 at 0x6dd: file demo.c, line 11.
(gdb) r
Starting program: /home/cc/cs161/su21/staff/cs161-tad/demo/a.out

Breakpoint 1, main () at demo.c:10
(gdb) 
```



jinanjiang — ssh cs161-tad@hive17.cs.berkeley.edu — 118×42

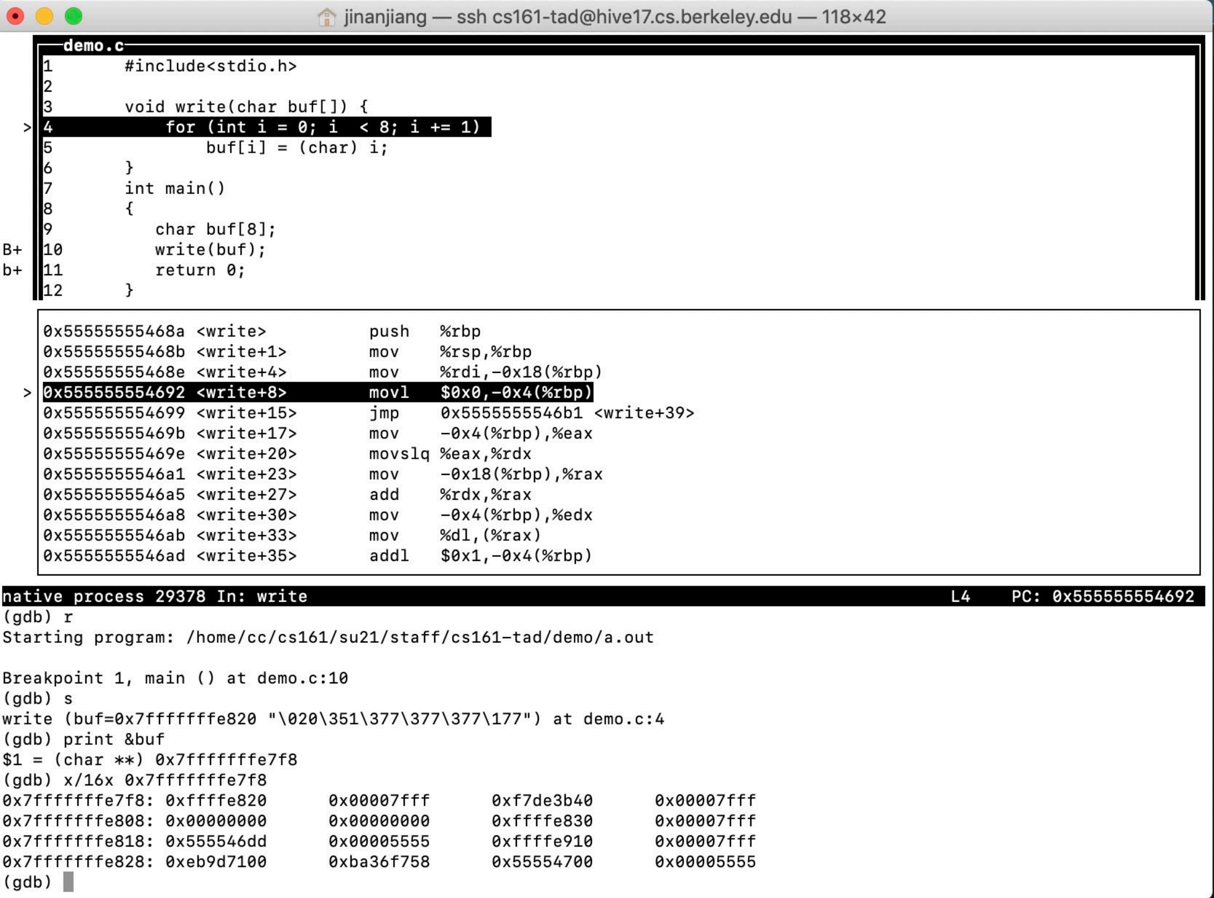
```
demo.c
    1       #include<stdio.h>
    2
    3       void write(char buf[]) {
>   4           for (int i = 0; i  < 8; i += 1)
    5               buf[i] = (char) i;
    6       }
    7       int main()
    8       {
    9          char buf[8];
B+  10         write(buf);
b+  11         return 0;
    12      }
```

```
  0x55555555468a <write>          push   %rbp
  0x55555555468b <write+1>        mov    %rsp,%rbp
  0x55555555468e <write+4>        mov    %rdi,-0x18(%rbp)
> 0x555555554692 <write+8>        movl   $0x0,-0x4(%rbp)
  0x555555554699 <write+15>       jmp    0x5555555546b1 <write+39>
  0x55555555469b <write+17>       mov    -0x4(%rbp),%eax
  0x55555555469e <write+20>       movslq %eax,%rdx
  0x5555555546a1 <write+23>       mov    -0x18(%rbp),%rax
  0x5555555546a5 <write+27>       add    %rdx,%rax
  0x5555555546a8 <write+30>       mov    -0x4(%rbp),%edx
  0x5555555546ab <write+33>       mov    %dl,(%rax)
  0x5555555546ad <write+35>       addl   $0x1,-0x4(%rbp)
```

```
native process 29378 In: write                                                     L4    PC: 0x555555554692
```

```
(gdb) 7 in /home/cc/cs161/su21/staff/cs161-tad/demo/demo.c
(gdb) b 10
Breakpoint 1 at 0x6d1: file demo.c, line 10.
(gdb) break 11
Breakpoint 2 at 0x6dd: file demo.c, line 11.
(gdb) r
Starting program: /home/cc/cs161/su21/staff/cs161-tad/demo/a.out

Breakpoint 1, main () at demo.c:10
(gdb) s
write (buf=0x7fffffffe820 "\020\351\377\377\377\177") at demo.c:4
(gdb) 
```



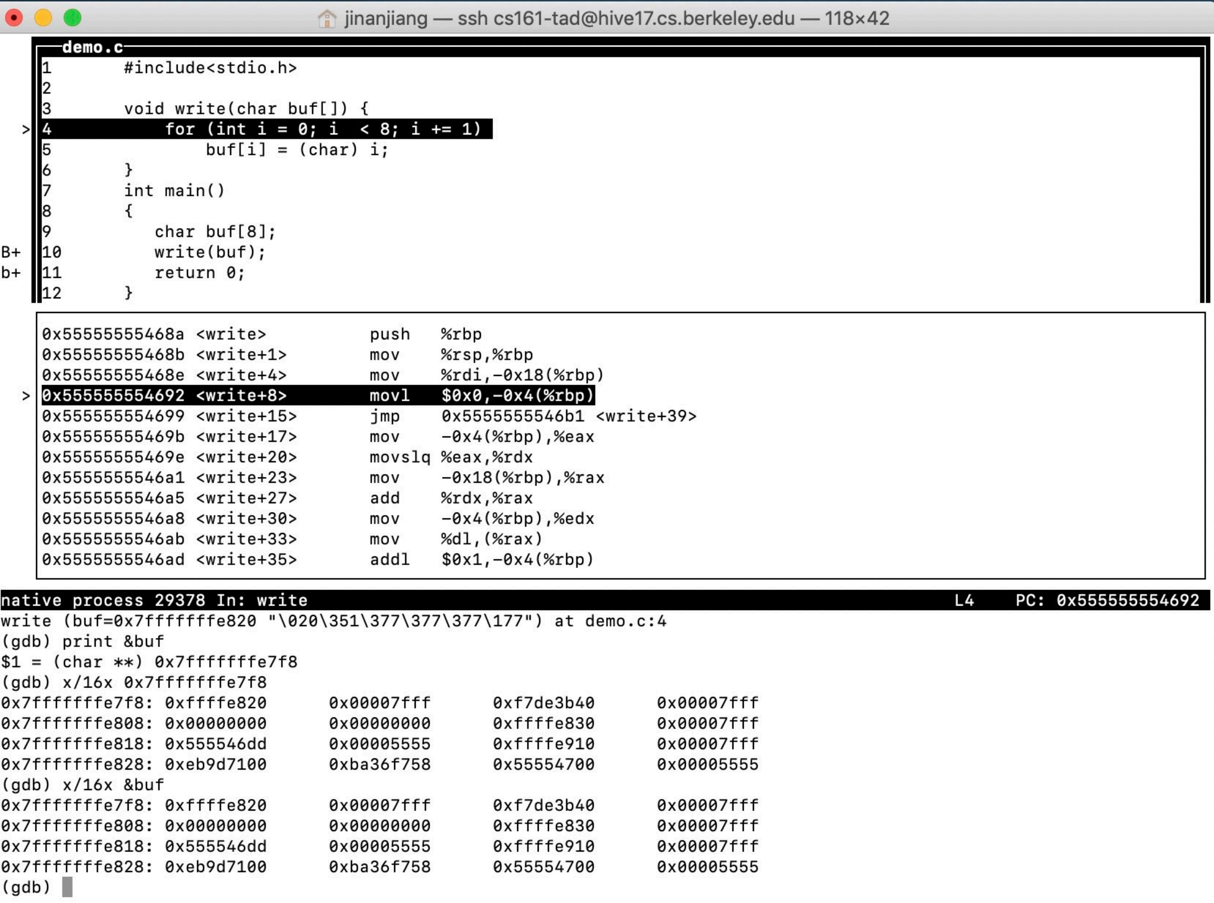
jinanjiang — ssh cs161-tad@hive17.cs.berkeley.edu — 118×42

```
demo.c
   1        #include<stdio.h>
   2
   3        void write(char buf[]) {
>  4            for (int i = 0; i  < 8; i += 1)
   5                buf[i] = (char) i;
   6        }
   7        int main()
   8        {
   9           char buf[8];
B+ 10          write(buf);
b+ 11          return 0;
   12       }
```

```
   0x55555555468a <write>          push   %rbp
   0x55555555468b <write+1>        mov    %rsp,%rbp
   0x55555555468e <write+4>        mov    %rdi,-0x18(%rbp)
>  0x555555554692 <write+8>        movl   $0x0,-0x4(%rbp)
   0x555555554699 <write+15>       jmp    0x5555555546b1 <write+39>
   0x55555555469b <write+17>       mov    -0x4(%rbp),%eax
   0x55555555469e <write+20>       movslq %eax,%rdx
   0x5555555546a1 <write+23>       mov    -0x18(%rbp),%rax
   0x5555555546a5 <write+27>       add    %rdx,%rax
   0x5555555546a8 <write+30>       mov    -0x4(%rbp),%edx
   0x5555555546ab <write+33>       mov    %dl,(%rax)
   0x5555555546ad <write+35>       addl   $0x1,-0x4(%rbp)
```

```
native process 29378 In: write                                                           L4    PC: 0x555555554692
(gdb) r
Starting program: /home/cc/cs161/su21/staff/cs161-tad/demo/a.out

Breakpoint 1, main () at demo.c:10
(gdb) s
write (buf=0x7fffffffe820 "\020\351\377\377\377\177") at demo.c:4
(gdb) print &buf
$1 = (char **) 0x7fffffffe7f8
(gdb) x/16x 0x7fffffffe7f8
0x7fffffffe7f8: 0xffffe820      0x00007fff      0xf7de3b40      0x00007fff
0x7fffffffe808: 0x00000000      0x00000000      0xffffe830      0x00007fff
0x7fffffffe818: 0x555546dd      0x00005555      0xffffe910      0x00007fff
0x7fffffffe828: 0xeb9d7100      0xba36f758      0x55554700      0x00005555
(gdb)
```

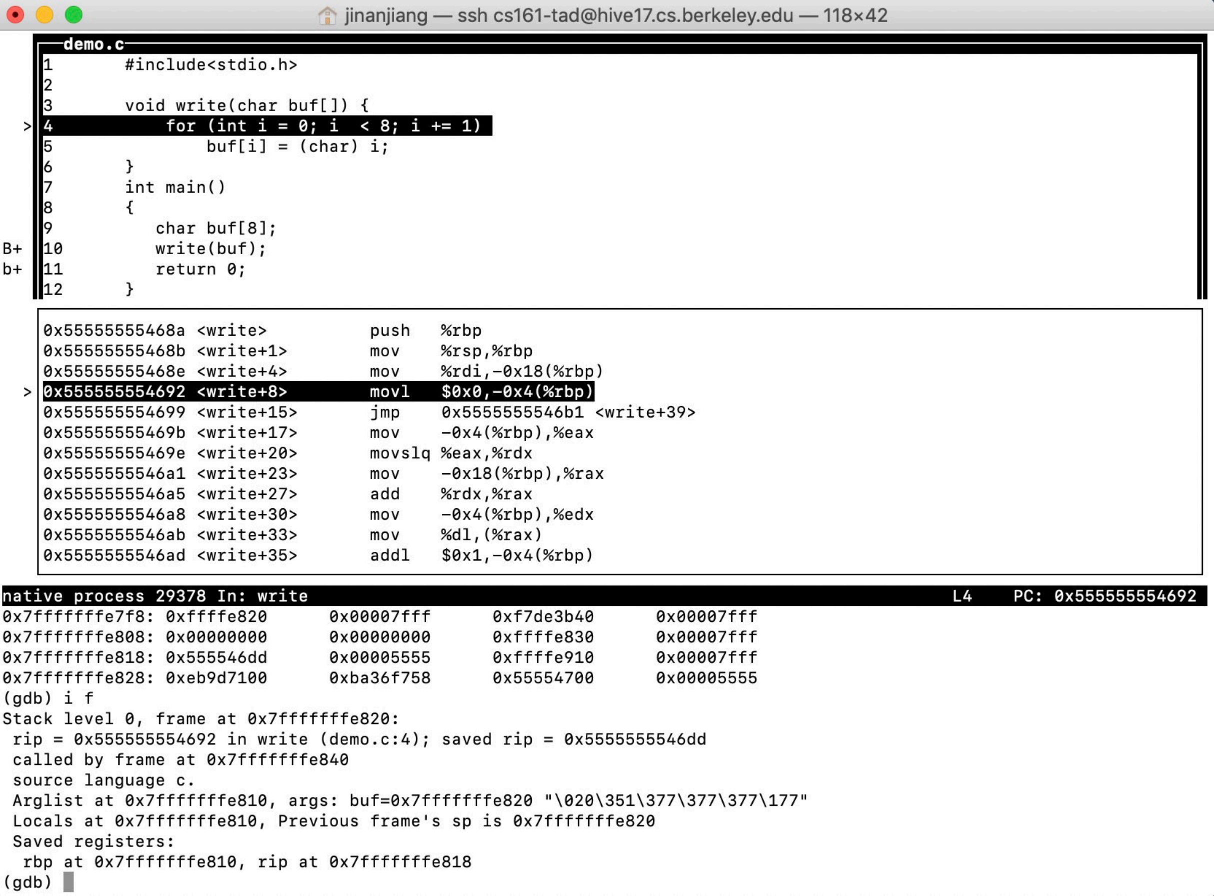


jinanjiang — ssh cs161-tad@hive17.cs.berkeley.edu — 118×42

```
   demo.c
   1       #include<stdio.h>
   2
   3       void write(char buf[]) {
 > 4           for (int i = 0; i  < 8; i += 1)
   5               buf[i] = (char) i;
   6       }
   7       int main()
   8       {
   9          char buf[8];
B+ 10         write(buf);
b+ 11         return 0;
   12      }
```

```
   0x55555555468a <write>          push   %rbp
   0x55555555468b <write+1>        mov    %rsp,%rbp
   0x55555555468e <write+4>        mov    %rdi,-0x18(%rbp)
 > 0x555555554692 <write+8>        movl   $0x0,-0x4(%rbp)
   0x555555554699 <write+15>       jmp    0x5555555546b1 <write+39>
   0x55555555469b <write+17>       mov    -0x4(%rbp),%eax
   0x55555555469e <write+20>       movslq %eax,%rdx
   0x5555555546a1 <write+23>       mov    -0x18(%rbp),%rax
   0x5555555546a5 <write+27>       add    %rdx,%rax
   0x5555555546a8 <write+30>       mov    -0x4(%rbp),%edx
   0x5555555546ab <write+33>       mov    %dl,(%rax)
   0x5555555546ad <write+35>       addl   $0x1,-0x4(%rbp)
```

```
native process 29378 In: write                                                              L4    PC: 0x555555554692
write (buf=0x7fffffffe820 "\020\351\377\377\377\177") at demo.c:4
(gdb) print &buf
$1 = (char **) 0x7fffffffe7f8
(gdb) x/16x 0x7fffffffe7f8
0x7fffffffe7f8: 0xffffe820      0x00007fff      0xf7de3b40      0x00007fff
0x7fffffffe808: 0x00000000      0x00000000      0xffffe830      0x00007fff
0x7fffffffe818: 0x555546dd      0x00005555      0xffffe910      0x00007fff
0x7fffffffe828: 0xeb9d7100      0xba36f758      0x55554700      0x00005555
(gdb) x/16x &buf
0x7fffffffe7f8: 0xffffe820      0x00007fff      0xf7de3b40      0x00007fff
0x7fffffffe808: 0x00000000      0x00000000      0xffffe830      0x00007fff
0x7fffffffe818: 0x555546dd      0x00005555      0xffffe910      0x00007fff
0x7fffffffe828: 0xeb9d7100      0xba36f758      0x55554700      0x00005555
(gdb) 
```

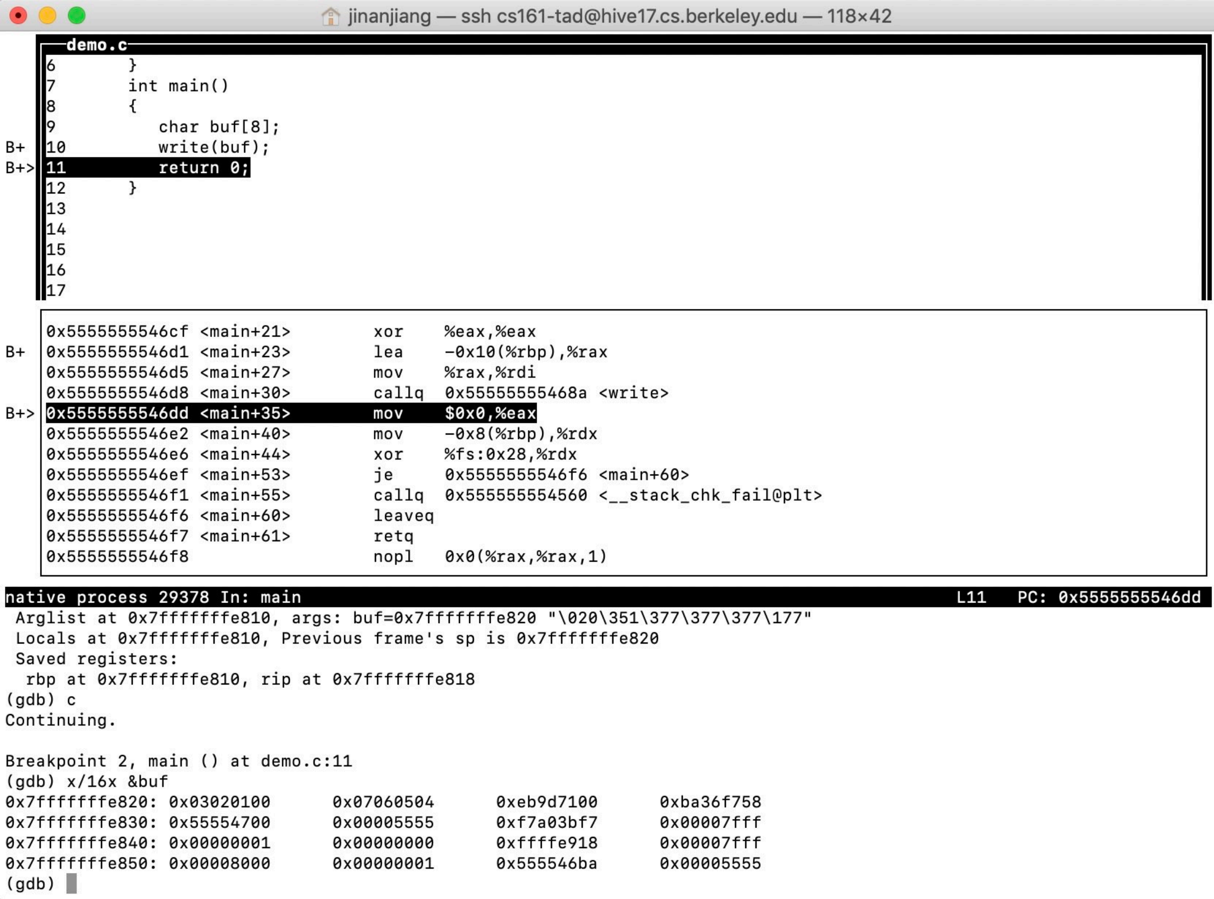


```
jinanjiang — ssh cs161-tad@hive17.cs.berkeley.edu — 118×42
```

```
   demo.c
   1        #include<stdio.h>
   2
   3        void write(char buf[]) {
>  4            for (int i = 0; i  < 8; i += 1)
   5                buf[i] = (char) i;
   6        }
   7        int main()
   8        {
   9           char buf[8];
B+ 10          write(buf);
b+ 11          return 0;
   12       }
```

```
   0x55555555468a <write>          push   %rbp
   0x55555555468b <write+1>        mov    %rsp,%rbp
   0x55555555468e <write+4>        mov    %rdi,-0x18(%rbp)
>  0x555555554692 <write+8>        movl   $0x0,-0x4(%rbp)
   0x555555554699 <write+15>       jmp    0x5555555546b1 <write+39>
   0x55555555469b <write+17>       mov    -0x4(%rbp),%eax
   0x55555555469e <write+20>       movslq %eax,%rdx
   0x5555555546a1 <write+23>       mov    -0x18(%rbp),%rax
   0x5555555546a5 <write+27>       add    %rdx,%rax
   0x5555555546a8 <write+30>       mov    -0x4(%rbp),%edx
   0x5555555546ab <write+33>       mov    %dl,(%rax)
   0x5555555546ad <write+35>       addl   $0x1,-0x4(%rbp)
```

```
native process 29378 In: write                                                     L4    PC: 0x555555554692
0x7fffffffe7f8: 0xffffe820      0x00007fff      0xf7de3b40      0x00007fff
0x7fffffffe808: 0x00000000      0x00000000      0xffffe830      0x00007fff
0x7fffffffe818: 0x555546dd      0x00005555      0xffffe910      0x00007fff
0x7fffffffe828: 0xeb9d7100      0xba36f758      0x55554700      0x00005555
(gdb) i f
Stack level 0, frame at 0x7fffffffe820:
 rip = 0x555555554692 in write (demo.c:4); saved rip = 0x5555555546dd
 called by frame at 0x7fffffffe840
 source language c.
 Arglist at 0x7fffffffe810, args: buf=0x7fffffffe820 "\020\351\377\377\377\177"
 Locals at 0x7fffffffe810, Previous frame's sp is 0x7fffffffe820
 Saved registers:
  rbp at 0x7fffffffe810, rip at 0x7fffffffe818
(gdb) 
```



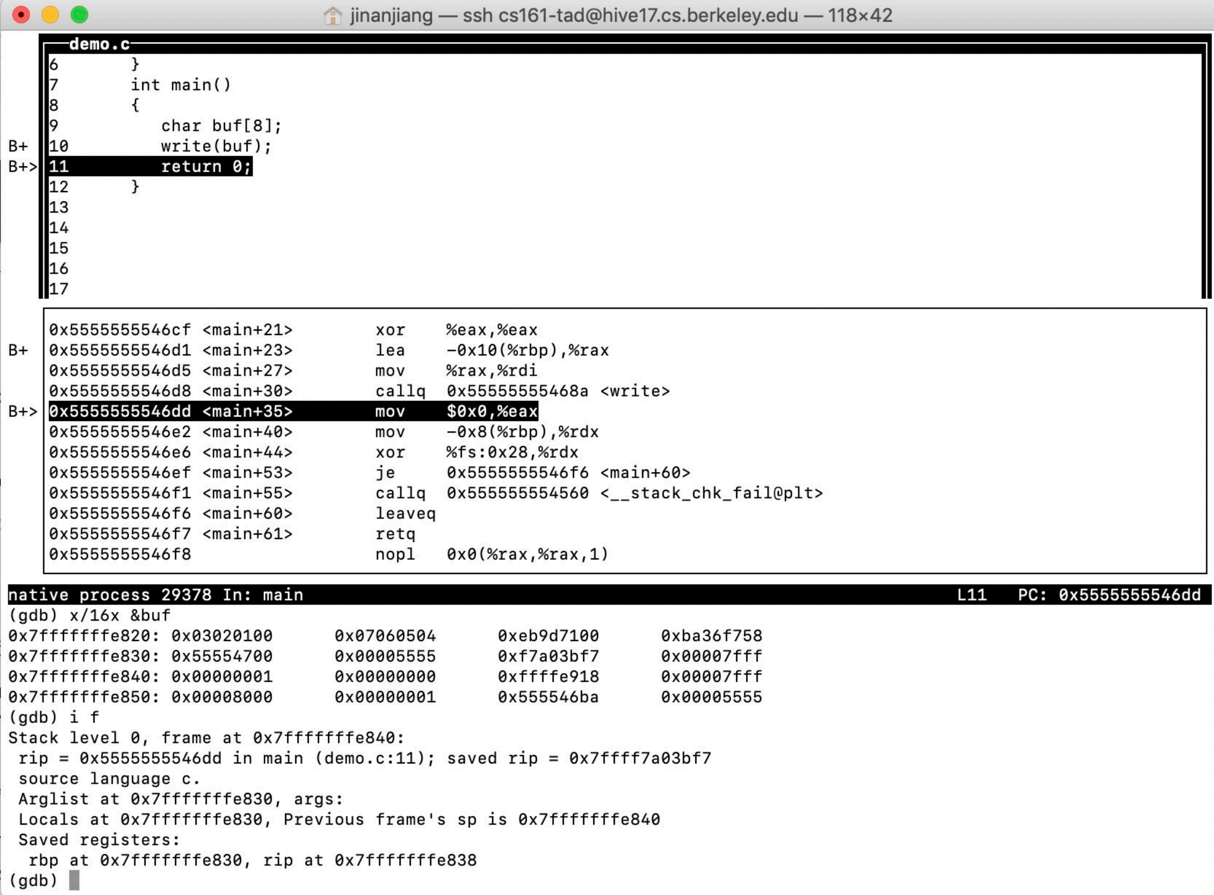
```
jinanjiang — ssh cs161-tad@hive17.cs.berkeley.edu — 118×42
```

```
   demo.c
   6        }
   7        int main()
   8        {
   9           char buf[8];
B+ 10          write(buf);
B+>11          return 0;
   12       }
   13
   14
   15
   16
   17
```

```
   0x5555555546cf <main+21>        xor    %eax,%eax
B+ 0x5555555546d1 <main+23>        lea    -0x10(%rbp),%rax
   0x5555555546d5 <main+27>        mov    %rax,%rdi
   0x5555555546d8 <main+30>        callq  0x55555555468a <write>
B+>0x5555555546dd <main+35>        mov    $0x0,%eax
   0x5555555546e2 <main+40>        mov    -0x8(%rbp),%rdx
   0x5555555546e6 <main+44>        xor    %fs:0x28,%rdx
   0x5555555546ef <main+53>        je     0x5555555546f6 <main+60>
   0x5555555546f1 <main+55>        callq  0x555555554560 <__stack_chk_fail@plt>
   0x5555555546f6 <main+60>        leaveq
   0x5555555546f7 <main+61>        retq
   0x5555555546f8                  nopl   0x0(%rax,%rax,1)
```

```
native process 29378 In: main                                                      L11   PC: 0x5555555546dd
 Arglist at 0x7fffffffe810, args: buf=0x7fffffffe820 "\020\351\377\377\377\177"
 Locals at 0x7fffffffe810, Previous frame's sp is 0x7fffffffe820
 Saved registers:
  rbp at 0x7fffffffe810, rip at 0x7fffffffe818
(gdb) c
Continuing.

Breakpoint 2, main () at demo.c:11
(gdb) x/16x &buf
0x7fffffffe820: 0x03020100      0x07060504      0xeb9d7100      0xba36f758
0x7fffffffe830: 0x55554700      0x00005555      0xf7a03bf7      0x00007fff
0x7fffffffe840: 0x00000001      0x00000000      0xffffe918      0x00007fff
0x7fffffffe850: 0x00008000      0x00000001      0x555546ba      0x00005555
(gdb) 
```

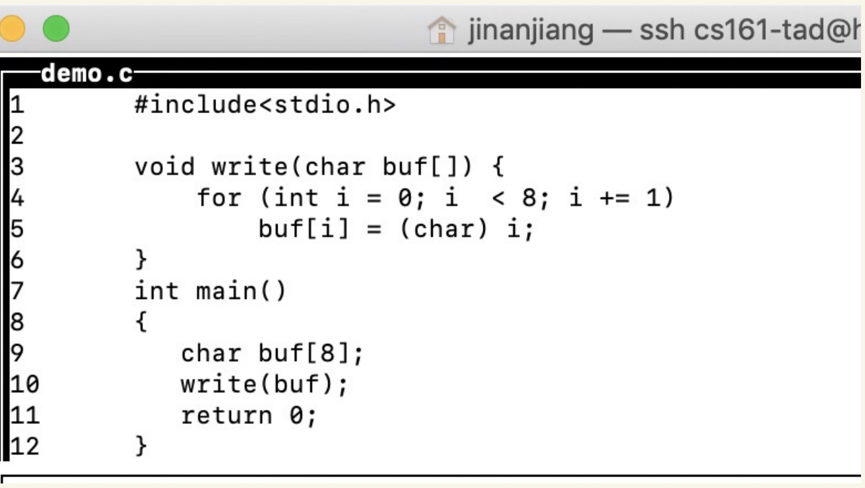


jinanjiang — ssh cs161-tad@hive17.cs.berkeley.edu — 118×42

```
   ┌─demo.c──────────────────────────────────────────────────────────
   │6        }
   │7        int main()
   │8        {
   │9            char buf[8];
B+ │10           write(buf);
B+>│11           return 0;
   │12       }
   │13
   │14
   │15
   │16
   │17
```

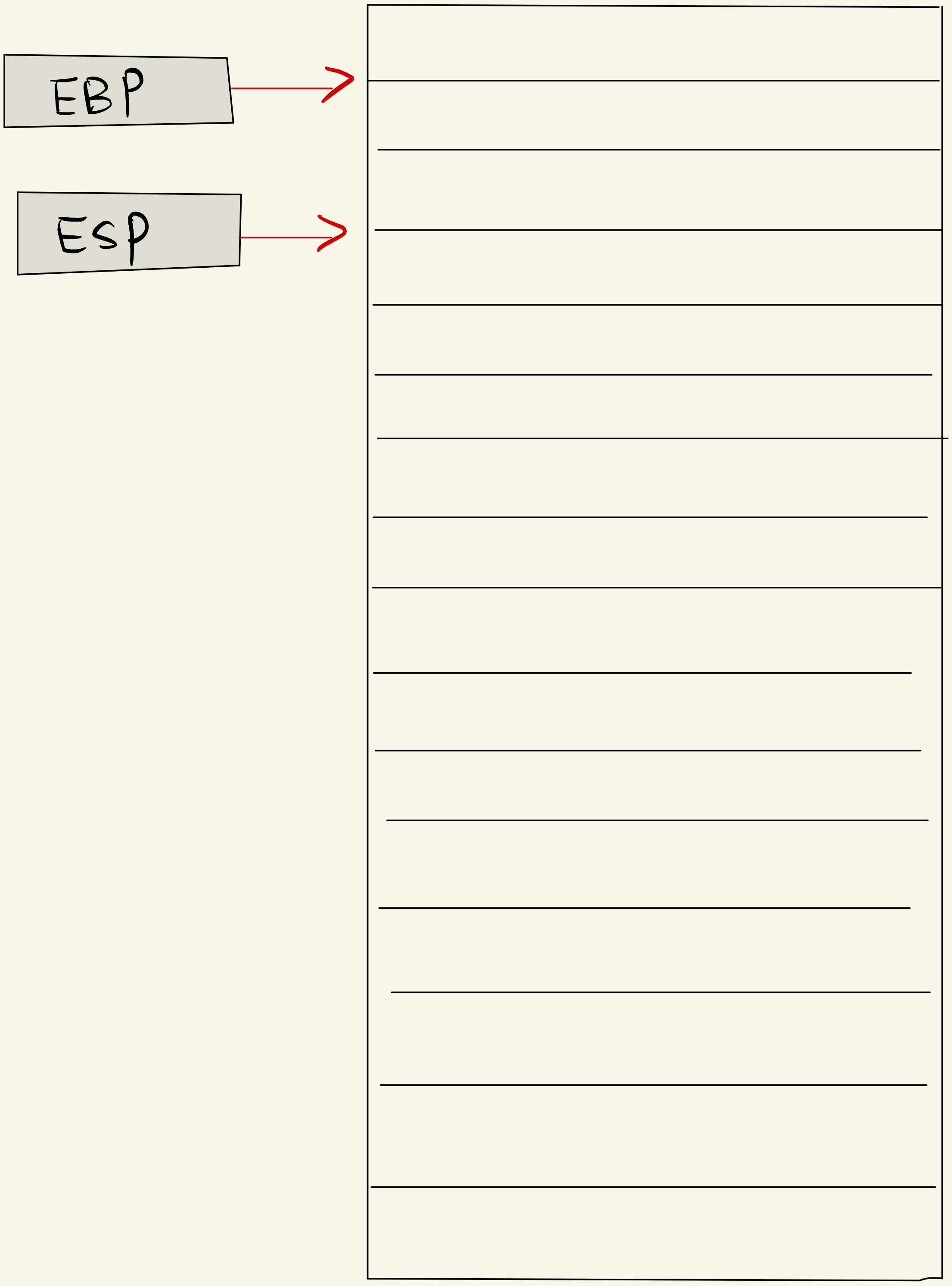
```
    0x5555555546cf <main+21>        xor    %eax,%eax
B+  0x5555555546d1 <main+23>        lea    -0x10(%rbp),%rax
    0x5555555546d5 <main+27>        mov    %rax,%rdi
    0x5555555546d8 <main+30>        callq  0x55555555468a <write>
B+> 0x5555555546dd <main+35>        mov    $0x0,%eax
    0x5555555546e2 <main+40>        mov    -0x8(%rbp),%rdx
    0x5555555546e6 <main+44>        xor    %fs:0x28,%rdx
    0x5555555546ef <main+53>        je     0x5555555546f6 <main+60>
    0x5555555546f1 <main+55>        callq  0x555555554560 <__stack_chk_fail@plt>
    0x5555555546f6 <main+60>        leaveq
    0x5555555546f7 <main+61>        retq
    0x5555555546f8                  nopl   0x0(%rax,%rax,1)
```

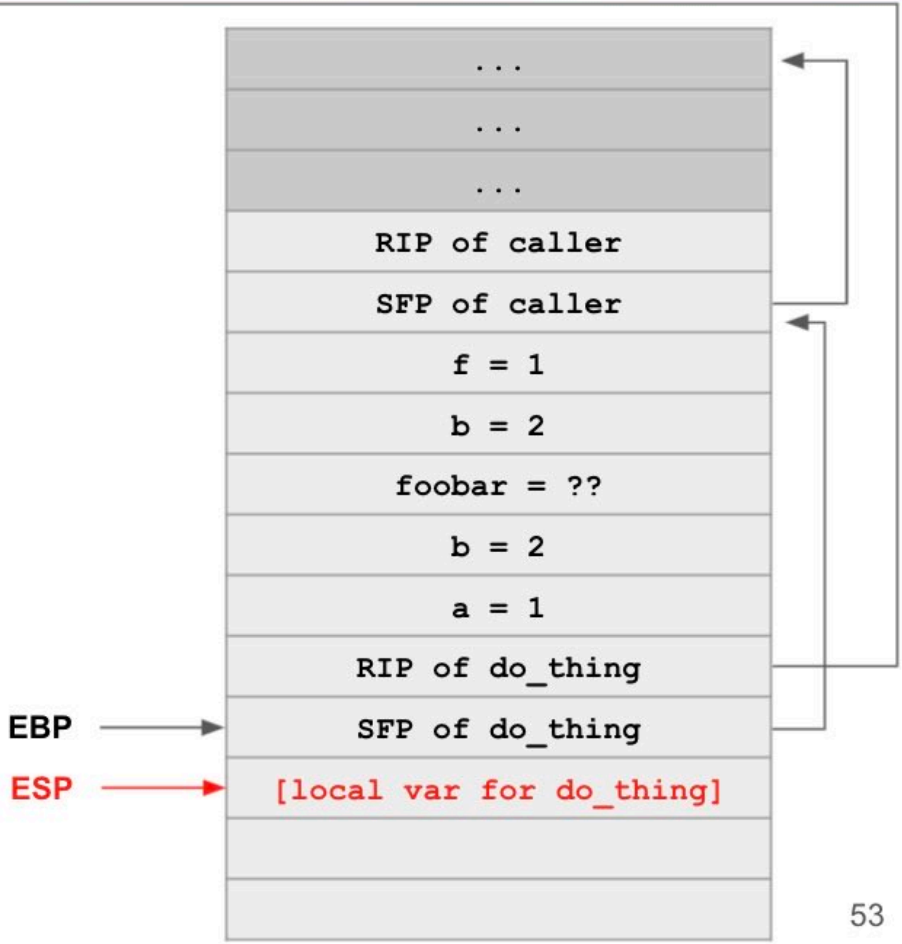
```
native process 29378 In: main                                                     L11   PC: 0x5555555546dd
(gdb) x/16x &buf
0x7fffffffe820: 0x03020100      0x07060504      0xeb9d7100      0xba36f758
0x7fffffffe830: 0x55554700      0x00005555      0xf7a03bf7      0x00007fff
0x7fffffffe840: 0x00000001      0x00000000      0xffffe918      0x00007fff
0x7fffffffe850: 0x00008000      0x00000001      0x555546ba      0x00005555
(gdb) i f
Stack level 0, frame at 0x7fffffffe840:
 rip = 0x5555555546dd in main (demo.c:11); saved rip = 0x7ffff7a03bf7
 source language c.
 Arglist at 0x7fffffffe830, args:
 Locals at 0x7fffffffe830, Previous frame's sp is 0x7fffffffe840
 Saved registers:
  rbp at 0x7fffffffe830, rip at 0x7fffffffe838
(gdb) 
```



```c
#include<stdio.h>

void write(char buf[]) {
    for (int i = 0; i  < 8; i += 1)
        buf[i] = (char) i;
}
int main()
{
   char buf[8];
   write(buf);
   return 0;
}
```




...
...
...
RIP of caller
SFP of caller
f = 1
b = 2
foobar = ??
b = 2
a = 1
RIP of do_thing
SFP of do_thing
[local var for do_thing]
EBP
ESP