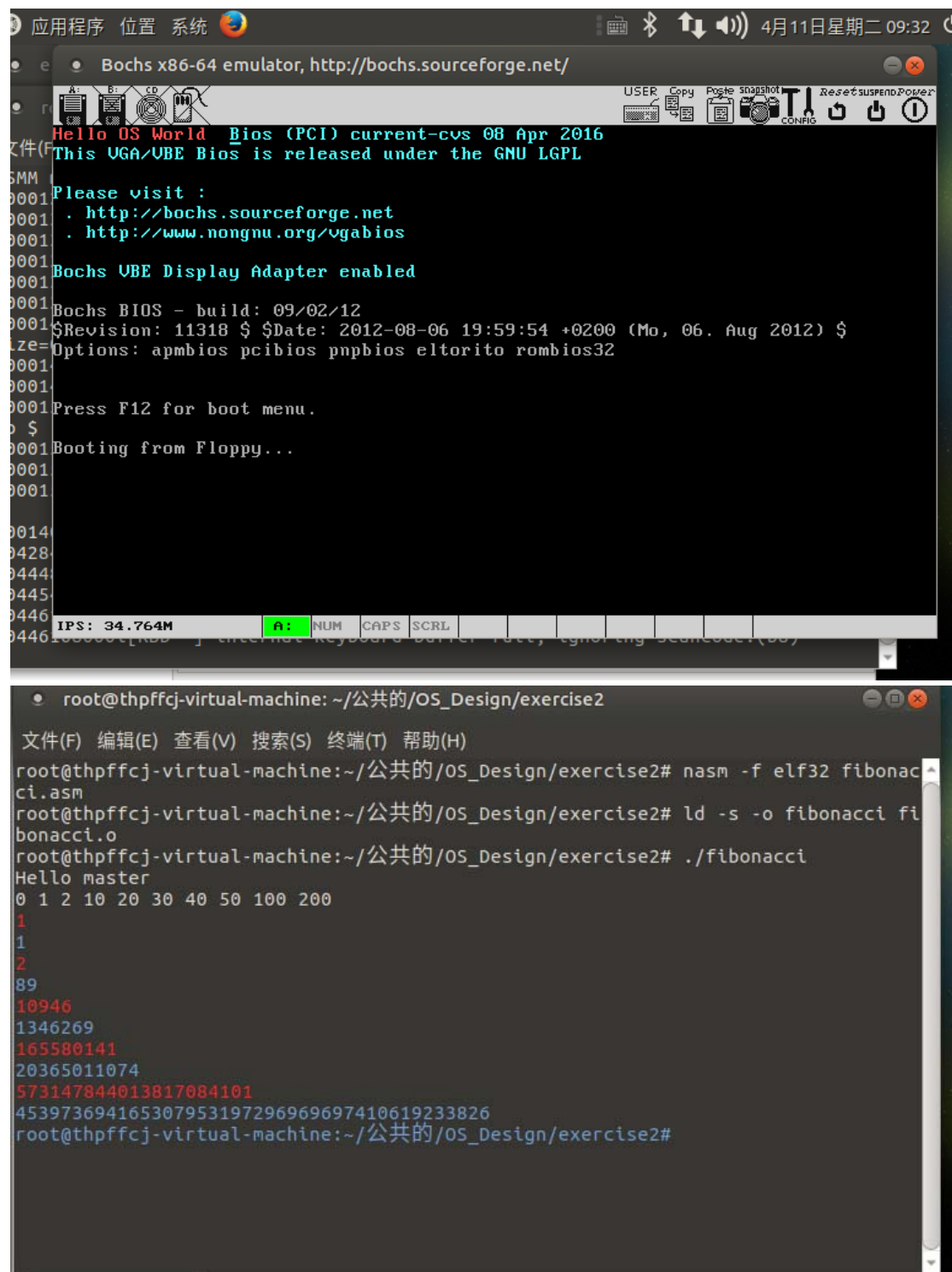


运行环境 ubuntu-mate-16.10-32 32 位



```
Bochs x86-64 emulator, http://bochs.sourceforge.net/
Hello OS World Bios (PCI) current-cvs 08 Apr 2016
This VGA/VE BIOS is released under the GNU LGPL
Please visit :
. http://bochs.sourceforge.net
. http://www.nongnu.org/vgabios
Bochs VBE Display Adapter enabled
Bochs BIOS - build: 09/02/12
Revision: 11318 $ $Date: 2012-08-06 19:59:54 +0200 (Mo, 06. Aug 2012) $
Options: apmbios pcibios pnpbios eltorito rombios32
Press F12 for boot menu.
Booting from Floppy...
IPS: 34.764M
```

```
root@thpffcj-virtual-machine: ~/公共的/OS_Design/exercise2
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
root@thpffcj-virtual-machine:~/公共的/OS_Design/exercise2# nasm -f elf32 fibonacci.asm
root@thpffcj-virtual-machine:~/公共的/OS_Design/exercise2# ld -s -o fibonacci fibonacci.o
root@thpffcj-virtual-machine:~/公共的/OS_Design/exercise2# ./fibonacci
Hello master
0 1 2 10 20 30 40 50 100 200
1
1
2
89
10946
1346269
165580141
20365011074
573147844013817084101
453973694165307953197296969697410619233826
root@thpffcj-virtual-machine:~/公共的/OS_Design/exercise2#
```

遇到问题及解决

除去大数计算，主要的问题是开始的时候用了栈，导致计算和输出的结果会

与输入结果相反。问了一些同学，他们没有用栈，输入一个计算一个输出一个。但不想改了，就用了一个比较笨的方法，题目中说了输入个数 1-9，就申请了 10 个内存读出来之后再进行一次压栈来调整顺序。

解决大数计算

和以前写的大数计算差不多，我在计算时基本使用 8 位寄存器，最多可以表示三位数，所以采取满 100 进 1 的策略，申请了 32 bytes 来计算结果，从后向前计算，满 100 进 1，当前两位减去 100。计算下两位后加上进位，进位归 0，循环下去。一共可以表示 64 位结果。

对于下面两行代码：

```
mov ax, BootMessage
```

```
mov bp, ax
```

思考为什么 `mov bp, ax` 后，`int 10h` 就能够取到 `BootMessage` 了？运行到这行代码的时候 `ax` 里面的值是多少？这个值是不是 `BootMessage` 所在内存中的位置（即相对地址还是绝对地址）？

10H 中断用于显示字符串，`bp` 指向要显示的字符串，调用中断就显示了字符串。

`ax` 中的值是 `BootMessage` 对应的字符串的首地址。这个值是 `BootMessage` 的绝对地址。