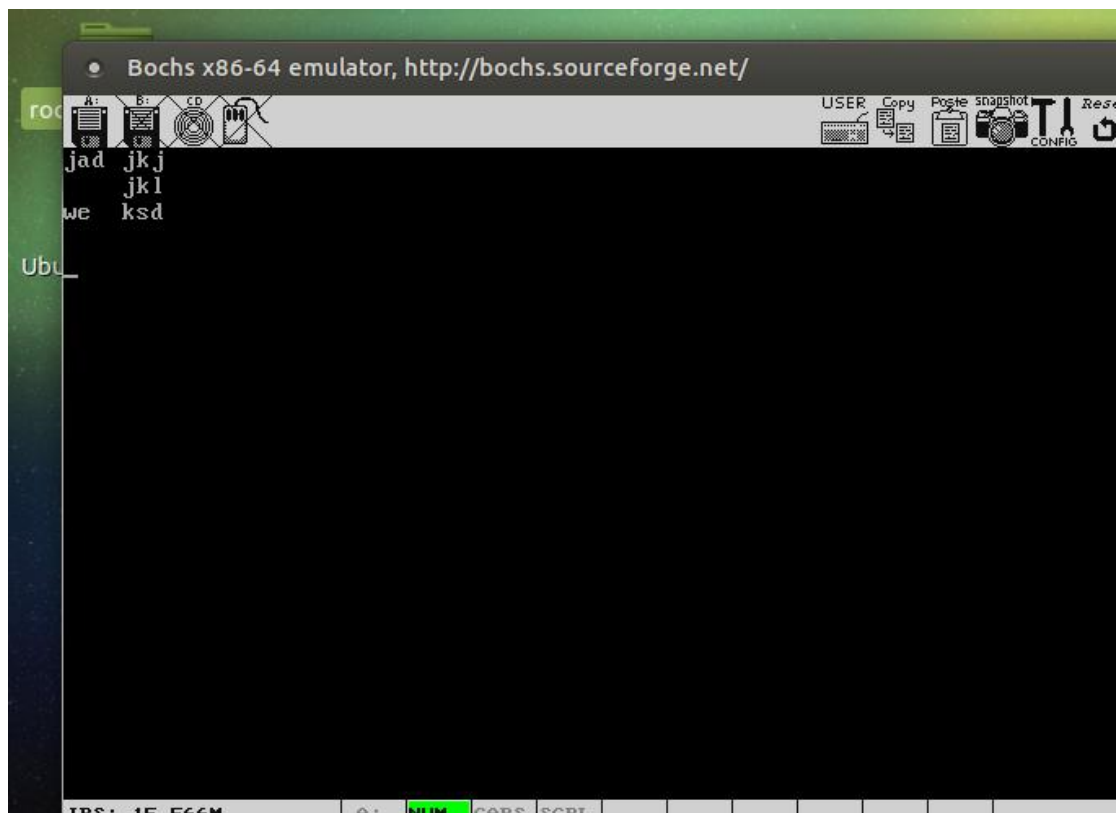
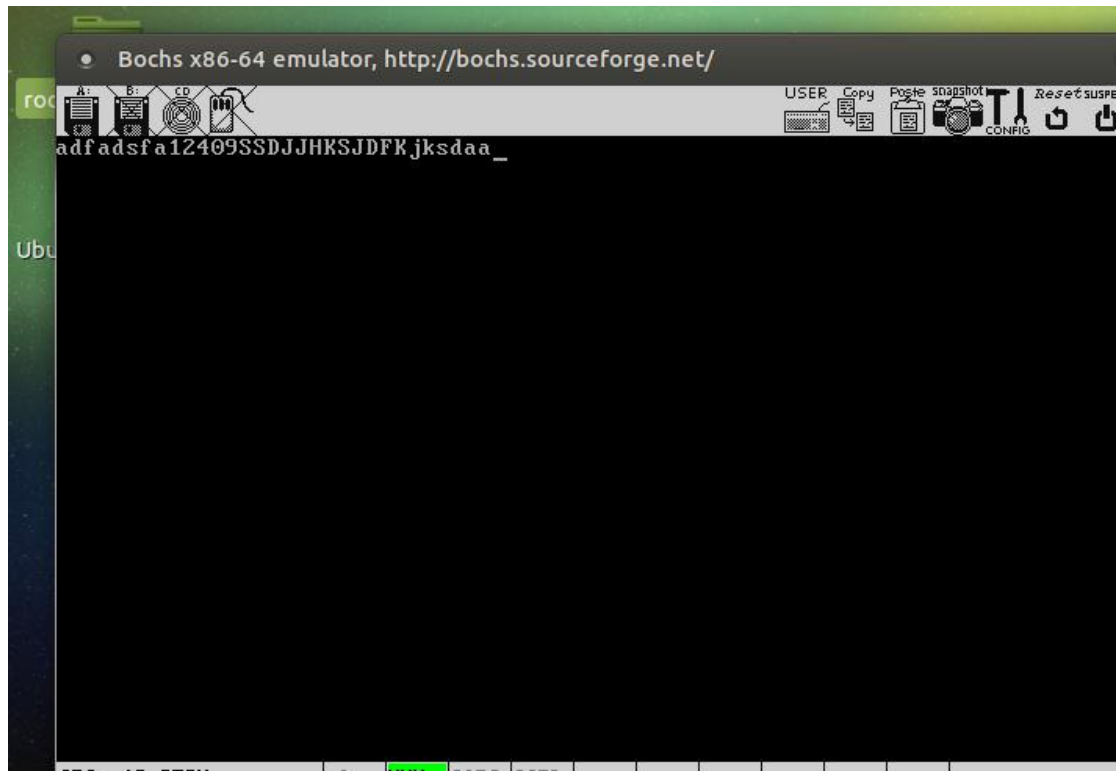
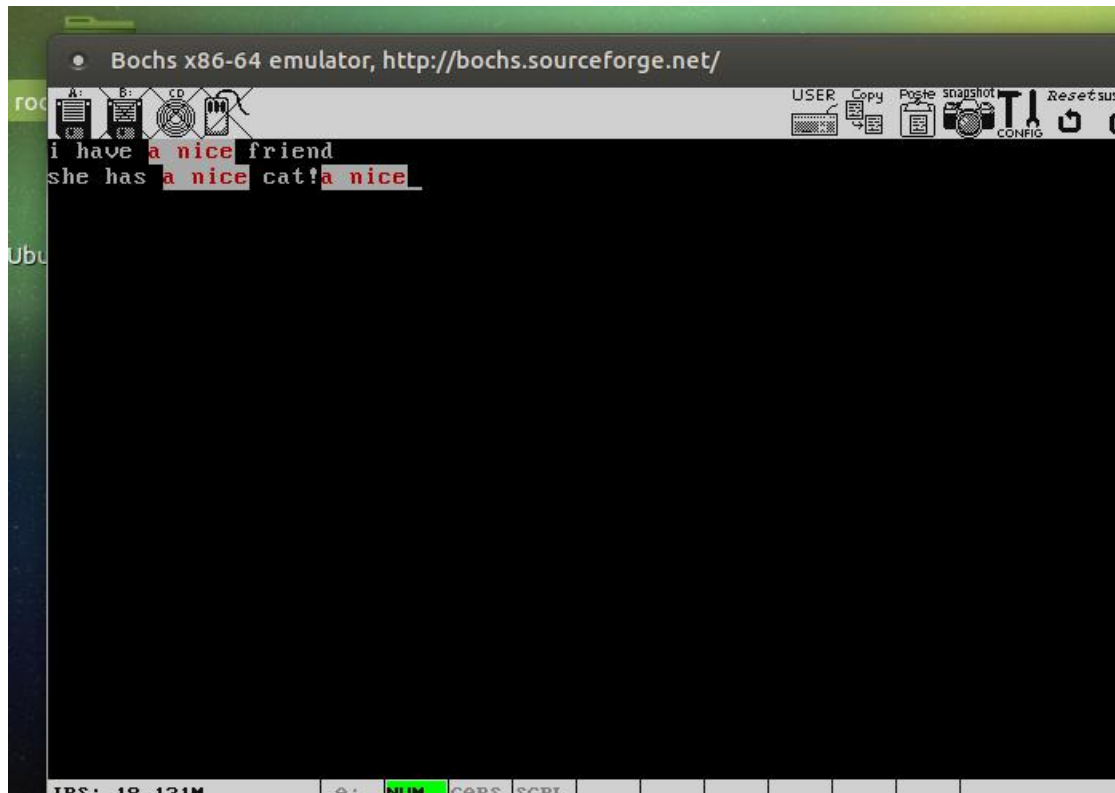


运行环境 ubuntu-mate-16.10-32 位





第一张图展示字符输入包括大小写，第二张图展示 `tab` 的对其效果，第三张图展示搜索。

这次实验我是在 Orange S 第七章 m 的基础上改写代码，主要改写了 `console.c` 和 `tty.c`，添加了 `seek.c` 用来查找相同的字符串。

主要过程是当 TTY 任务开始运行时，所有 TTY 都将被初始化，并且全局变量 `nr_current_console` 会被赋值为 0，然后循环开始并一直进行下去。对于每一个 TTY，首先执行 `tty_do_read()`，他将调用 `key_board_read()`，并将读入的字符交给函数 `in_process()` 来处理，如果是需要输出的字符，会被 `in_process()` 放入当前接受处理的 TTY 的缓冲区中。然后 `tty_do_write()` 会接着执行，如果缓冲区中有数据，就被送入 `out_char()` 显示出来。

实现搜索是在 `CONSOLE` 这个数据结构里添加数组记录

- 1 输出过的所有字符
- 2 这个字符被输出之前光标的位置
- 3 输出过的所有字符对应的颜色
- 4 被查找的目标字符串缓存与当前位置

调用自己写的 `seek` 类中判断相等方法，传入输入字符数组和要查询的字符数组进行搜索，返回查到的字符位置，改变这个位置字符的颜色实现搜索模式。

实现 `tab` 主要靠模 4 确定需要输出的空格数目，删除时因为 `CONSOLE` 记录了这个字符被输出之前光标的位置就可以实现整体删除。

实现 20 秒刷新是调用 `syscall.asm` 中的 `getget_ticks()` 方法