

项目设计与功能说明文档

小组成员：

陈子杰 1850106

马天放 1750773

黎力 1853549

黄伟杰 1850107

霍定镭 1850110

指导教师：王冬青

提交时间：2020-08-23

项目概述

项目目的

在经过一学期的操作系统课程学习之后，我们对操作系统的各个模块，如进程、内存、文件系统等有了较为清晰的了解，课程中的小项目也让我们大致了解了其代码实现。但是，我们对于这些模块如何整合在一起，共同运作并最终共同构成一个文件系统还有一定的疑问。因此，本项目的目的是让我们跟着书籍一起实现一个简单的操作系统，真正了解如何将课本上的知识整合成一个操作系统，并发挥自己的创造力，将一些自己的创新点加入操作系统中。

开发环境

a.基于32位Ubuntu开源GNU/Linux操作系统

b.使用Bochs开源模拟器

项目指标

本门操作系统课程设计中，5人为一组，利用《Orange's 一个操作系统的实现》书中提供的源代码，通过修改或者重新实现参考源码的一个或多个模块来实现一个简单的操作系统。

本系统主要针对文件系统进行重新实现，其中新增代码量达到文件模块代码的一半，主要增加了文件夹的隐藏功能，并将源代码的扁平状文件系统修改为树状文件系统实现**B级**项目难度；在参考源码的基础上实现系统级应用，如磁盘，工具台等，通过调用较多的系统API以实现对系统的检测和控制，实现**C级**项目难度；同时实现了用户级应用，实现**D级**项目难度。

设计方案

概述

本系统参考《Orange’s 一个操作系统的实现》书中提供的源代码，重新实现了文件系统部分，可以对文件/目录进行增删改写操作，同时实现了多级目录，可在不同层级目录中进行转换，并实现了文件夹的隐藏、上锁功能；同时通过调用部分系统API实现一些系统小游戏，并且增加了特别的开机动画。

系统功能结构及使用说明

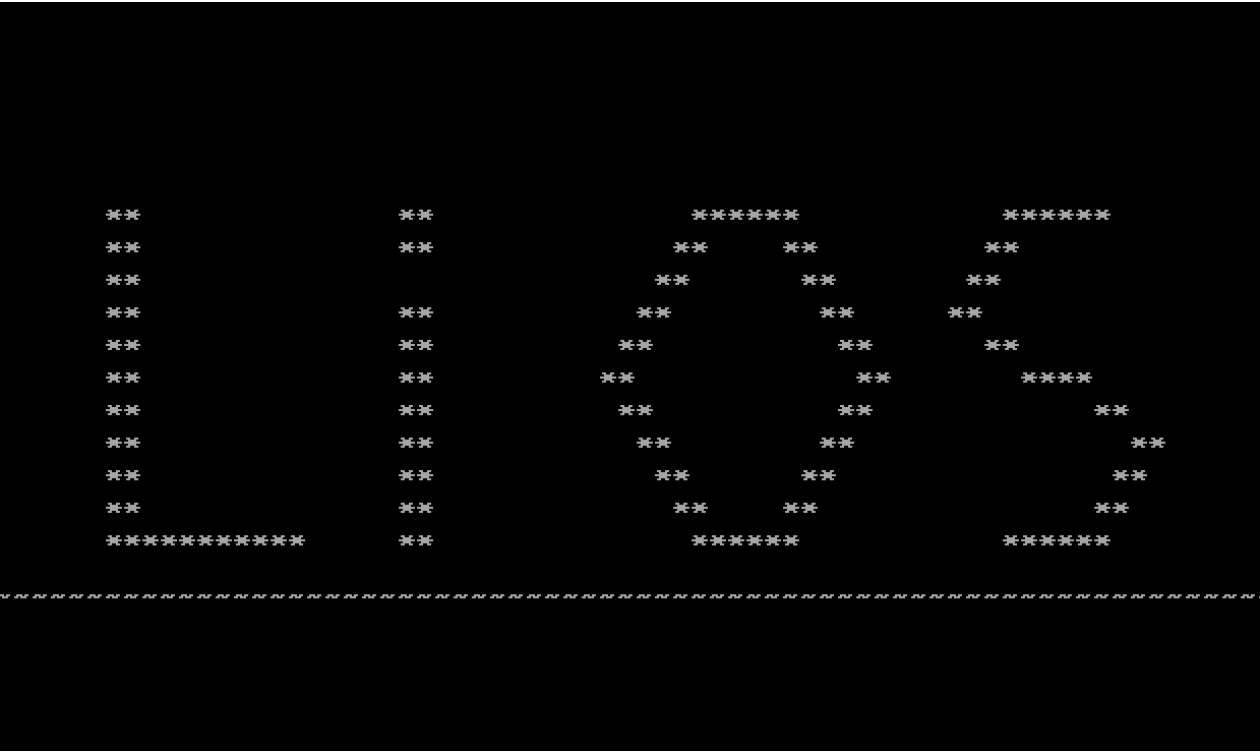
开机动画

◆ 描述

进入系统后，代表本操作系统名称的“LiOS”字样从屏幕四边进入，随后按顺时针淡出界面，3秒后自动消失。

◆ 实现方法

开机动画使用逐帧printf的方式实现，通过设置较小的延迟时间形成动画效果。



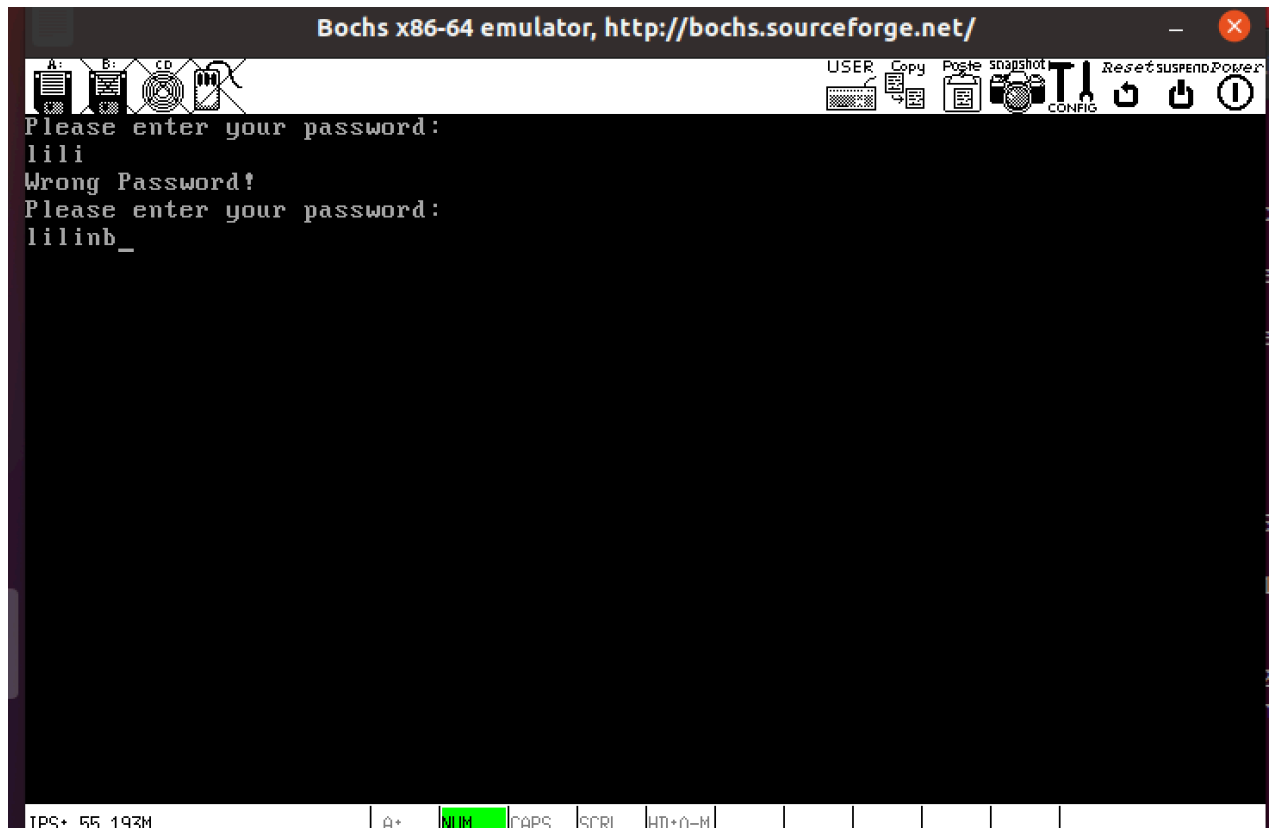
用户登录

◆ 描述

开机动画结束后，屏幕打印“Please enter your password!”，用户输入密码，密码正确则进入程序。如多次密码输入错误，则需等待一段时间才能再次输入密码。

◆ 实现方法

使用一个变量记录设置的密码，并在用户输入密码后进行比对。如多次错误，则在循环中增加delay_time，以增加等待时间。



帮助页面

◆ 描述

开机动画结束后，系统自动显示欢迎界面，用户输入“help”指令后，显示当前指令列表，方便用户进行相应操作；指令输入区“[root@localhost: /]”显示当前路径，“/”表示当前所在目录为根目录。

◆ 实现方法

用户输入指令后，系统通过判断字符串调用对应API，进入相应功能（详见下文）。

```

[root@LiOS: ~/help
=====LiOS help info=====
Command List
1. help : Show this help message
2. process : A process manage,show you all process-info here
3. clear : Clear the screen
4. ls : List all files in current directory
5. touch [filename] : Create a new file in current directory
6. rm [filename] : Delete a file in current directory
7. cat [filename] : Print the content of a file in current directory
8. vi [filename] : Write new content at the end of the file
9. mkdir [dirname] : Create a new directory in current directory
10. cd [dirname] : Go to a directory in current directory
11. hide [filename] : Hide a file
12. show [filename] : Unhide a file
13. game : Show the game list
14. time : Show the system current time
15. messageA/B/C : Send message to the terminal A or B or C
16. processA/B/C : Switch to the terminal A or B or C
17. calendar : A calendar application
18. time : Set the system time
=====
[root@LiOS: ~/

```

时钟功能

◆ 描述

用户输入“time”命令后，如果是第一次输入，则提示用户设置时间，第二次及以后则返回当前时间或者发出修改当前时间的指令

◆ 实现方法

采用testc()作为一个进程来专门运行时间函数，管理时间，确保不在时钟功能中时时钟也会照常运行。

```

=====
[welcome]
=====
[root@LiOS: ~/ltime
Please set current time! eg: 18:20:00
20:29:59
[root@LiOS: ~/ltime
Enter 'gettime' to get current time;
Enter a specific time point to change current time. eg:18:20:20
gettime
Current Time:
20:30:3
[root@LiOS: ~/ltime
Enter 'gettime' to get current time;
Enter a specific time point to change current time. eg:18:20:20
20:20:20
[root@LiOS: ~/ltime
Enter 'gettime' to get current time;
Enter a specific time point to change current time. eg:18:20:20
gettime
Current Time:
20:20:23
[root@LiOS: ~/

```

日历功能

◆ 描述

用户输入“calendar”命令后，提示用户输入正确的年份与月份，返回该月的日历。

◆ 实现方法

通过调用Calendar()函数计算当前年份是否为闰年，进一步计算日历的第一天是星期几，并按顺序输出日历。

```
[root@LiOS: ~]# calendar
Please enter year and month! eg:2020 08
2010 08
|=====|
|The Calendar of Year 2010 Month 8|
| 8  SUN MON TUE WED THU FRI SAT  |
|      1   2   3   4   5   6   7   |
|      8   9  10  11  12  13  14   |
|     15  16  17  18  19  20  21   |
|     22  23  24  25  26  27  28   |
|     29  30  31                   |
|=====|
[root@LiOS: ~]#
```

清屏功能

◆ 描述

用户输入“clear”命令后，系统进行清屏，显示上图初始欢迎界面。

◆ 实现方法

通过调用clear() API实现清屏功能，同时重新设置控制台相应初始值。



```
=====
LiOS v1.0.0
Kernel on Orange's

Welcome !
=====
root@LiOS: ~]#
```

The screenshot shows a terminal window with a top toolbar containing icons for USER, Copy, Paste, snapshot, CONFIG, Reset, suspend, and Power. The terminal content displays the LiOS v1.0.0 kernel welcome message. At the bottom, a status bar shows 'PS: 57.606M' and a row of function keys: A:, NUM, CAPS, SCRL, HD:0-M, and several empty slots.

文件管理功能

本文件功能使用树状文件系统，参考linux文件系统的指令进行指令设计，同时加入windows文件系统下的隐藏文件夹与文件上锁功能，

显示目录文件列表

◆ 描述

用户输入“ls”指令，系统打印出当前文件目录下的全部文件及目录。

通过调用ls()方法向文件系统发送LS系统消息，当文件系统接收到消息之后会执行do_ls()方法，该方法会通过遍历当前节点下的子节点来打印当前目录下的文件信息。打印的内容包括子节点的inode编号，对应的文件类型和对应的文件名。

```
[root@LiOS: /]ls
inode      type      filename
=====
1          file      .
2          file      dev_tty0
3          file      dev_tty1
4          file      dev_tty2
=====
[root@LiOS: /]
```

IPS: 53.837M | A: NUM | CAPS | SCRL | HD: 0-M | | | | |

创建文件

◆ 描述

用户输入“touch + filename”指令，在当前目录下创建一个新的文件并命名，若创建成功则系统提示“file created”的消息，若创建失败则系统提示“failed to create a new file”的消息。

◆ 实现方法

通过调用CreateFile()方法，该方法会调用open()方法来进行创建。根据方法返回值判断是否创建成功并给用户发出反馈消息。其中创建文件时会对文件进行重名检测。

```
[root@LiOS: /]cd ae
[root@LiOS: /ae/]touch 1
File created: 1 (fd 10)
[root@LiOS: /ae/]ls

inode      type      filename
=====
   11      file      1
=====
[root@LiOS: /ae/]
```

删除文件/文件夹

◆ 描述

用户输入“rm+filename”指令，删除指定文件名的文件或文件夹（由于不允许相同目录下文件重名，所以此处可以使用文件名进行搜索），删除成功后系统提示“deleted”，若删除失败（当前目录下不存在该文件）则会出现删除失败的信息。

◆ 实现方法

调用DeleteFile()方法，向文件系统发送UNLINK消息，文件系统在接收到消息之后会调用do_unlink()方法进而实现文件或文件夹的删除。其中文件夹的删除实现是递归删除文件夹里的子节点。

```

inode      type      filename
=====
    1      file      .
    2      file      dev_tty0
    3      file      dev_tty1
    4      file      dev_tty2
    7      [dir]      ae
=====
[root@LiOS: ~]rm ae
ae deleted!
[root@LiOS: ~]ls

inode      type      filename
=====
    1      file      .
    2      file      dev_tty0
    3      file      dev_tty1
    4      file      dev_tty2
=====
[root@LiOS: ~]

```

打印文件内容

◆ 描述

用户输入“cat+filename”指令，查看该文件内容，若文件中已写入内容，则如上图显示；若文件创建后未写入内容，则打印空行；若找不到文件则显示“fail to open”；若找到文件后打开失败则显示“An error has occurred in reading the file”。

◆ 实现方法

首先使用open()方法定位文件，然后调用read()方法对文件进行读操作。

```

[root@LiOS: ~]mkdir ae
creating directory /ae succeeded!
[root@LiOS: ~]cd ae
[root@LiOS: /ae/]touch a
File created: a (fd 5)
[root@LiOS: /ae/]vi a
asdfkasjhfd
[root@LiOS: /ae/]cat a
asdfkasjhfd
[root@LiOS: /ae/]

```

写文件

◆ 描述

用户输入“vi+filename”指令，系统在定位到目标文件后显示文件路径，之后用户输入要写入文件的内容并回车，完成写文件操作。

◆ 实现方法

首先使用open()方法定位文件，然后调用write()方法对文件进行写操作。

```
creating directory /ae succeeded!  
[root@LiOS: /]cd ae  
[root@LiOS: /ae/]touch a  
File created: a (fd 5)  
[root@LiOS: /ae/]vi a  
asdfkasjhfd  
[root@LiOS: /ae/]cat a  
asdfkasjhfd  
[root@LiOS: /ae/]
```

创建目录

◆ 描述

用户输入“mkdir + directoryname”指令，在当前目录内创建一个目录。

◆ 实现方法

首先使用CreateDir()方法创建对应文件路径，然后调用open()方法来检测是否有同名文件，如果有的话则会提示"Failed to create a new directory with name"。如果没有重名文件夹创建文件夹，成功或失败都会有相应提示信息。

```
Failed to create a new directory with name a  
[root@LiOS: /]mkdir ae  
creating directory /ae succeeded!  
[root@LiOS: /]ls  
  
inode      type      filename  
=====
```

1	file	.
2	file	dev_tty0
3	file	dev_tty1
4	file	dev_tty2
7	[dir]	a
11	[dir]	ae

```
=====
```

[root@LiOS: /]

进入多级目录

◆ 描述

用户输入“cd + directoryname”指令，系统自动检索目录路径，检索成功后，指令输入区的路径由“/”更改为“/directoryname/”，表示当前处于目标目录中。如果输入“cd ..”则会返回上一层。

◆ 实现方法

调用GoDir()方法，首先根据输入的目录名获得目标的绝对路径，并根据获得的绝对路径调用open()方法打开相应的文件。若想返回上一级，则会根据“/”找到目标位置，再将绝对路径修改为对应路径并更新；若想进入下一级，则将当前目标路径与目标文件夹名拼合，打开文件，直接修改路径并更新。

```
=====
[root@LiOS: /]cd ae
[root@LiOS: /ae/]ls

inode      type      filename
=====
     9      file      a

=====
[root@LiOS: /ae/]cd ..
[root@LiOS: /]ls

inode      type      filename
=====
     1      file      .
     2      file      dev_tty0
     3      file      dev_tty1
     4      file      dev_tty2
     7      [dir]     ae

=====
[root@LiOS: /]
```

隐藏文件夹

◆ 描述

用户输入“hide”指令后，程序调用“HideFile”方法找寻目标文件，若未找到目标文件，程序返回“File do not exist”；若找到了目标文件，则将该文件的状态置为“隐藏”。

◆ 实现方法

程序调用“HideFile”方法，在该方法中，先找寻用户希望隐藏的文件，然后将其状态置为“隐藏”，表示不希望该文件被系统识别。

```
File created: 1 (fd 10)
[root@LiOS: /ae/]ls

inode      type      filename
=====
   11      file      1
=====

[root@LiOS: /ae/]hide 1
Hide file 1 success!
[root@LiOS: /ae/]ls

inode      type      filename
=====
```

显示隐藏文件夹

◆ 描述

用户输入“show”指令后，程序调用“HideFile”方法找寻目标文件，若未找到目标文件，程序返回“File do not exist”；若找到了目标文件，则将该文件的状态置为“显示”。

◆ 实现方法

程序调用“HideFile”方法，在该方法中，先找寻用户希望隐藏的文件，然后将其状态置为“显示”，表示不希望该文件被系统识别。

```
-----
[root@LiOS: /ae/]show
Show file 1 success!
[root@LiOS: /ae/]ls

inode      type      filename
=====
   11      file      1
=====

[root@LiOS: /ae/]_
```

小游戏

本操作系统实现了一些系统小游戏功能。

扫雷

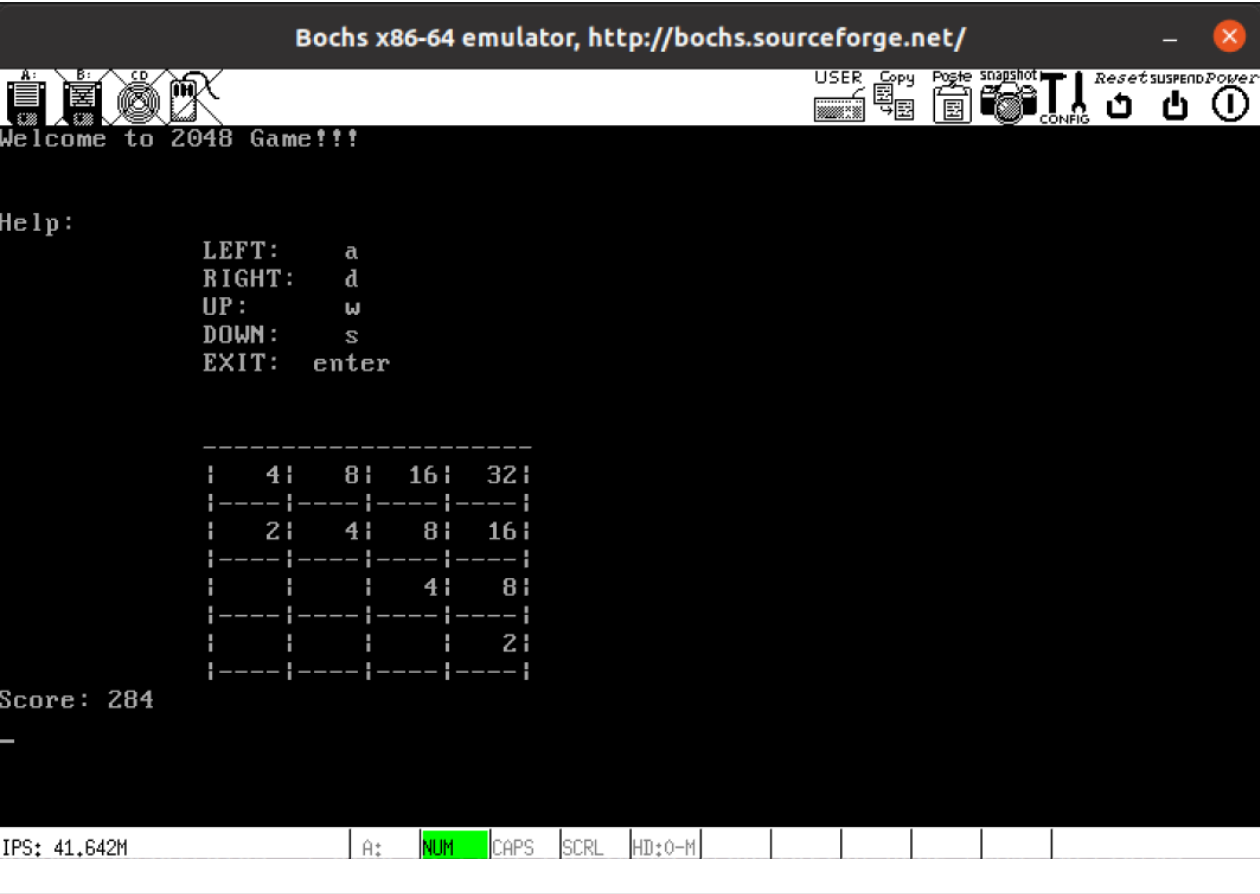
◆ 描述

◆ 描述

用户输入“game4”指令，进入2048游戏界面；系统自动打印游戏方法、游戏界面以及当前分数；用户每次只能输入一个运动方向，系统自动刷新棋盘及分数，同时在棋盘内随机生成新的数字。

◆ 实现方法

调用start2048Game()方法，其中调用多个2048相关函数，获取用户的键盘输入，共同实现2048的游戏操作。



进程系统

本操作系统实现了进程管理，进程间通信，进程切换等功能。

进程管理功能

◆ 描述

用户输入“process”指令，进入进程管理功能，系统打印出当前全部进程（包括系统进程和用户进程）的pid，名称，优先级（15表示系统级进程，5表示用户级进程）以及运行状态（YES表示正在运行，NO表示没有运行）。

◆ 实现方法

通过调用ProcessManage() API，遍历整个进程列表，并逐个打印。

```
18. time : Set the system time
=====
[root@LiOS: /]process
=====
pid      i      name      i      priority  i      running
0         TTY      15         YES
1         SYS      15         YES
2         HD       15         YES
3         FS       15         YES
4         MM       15         YES
5         TestA    5          YES
6         TestB    5          YES
7         TestC    5          YES
8         TestD    5          YES
=====
[root@LiOS: /]
IPS: 56.940M | A: NUM | CAPS | SCRL | HD:0-M | | | | | | | |
```

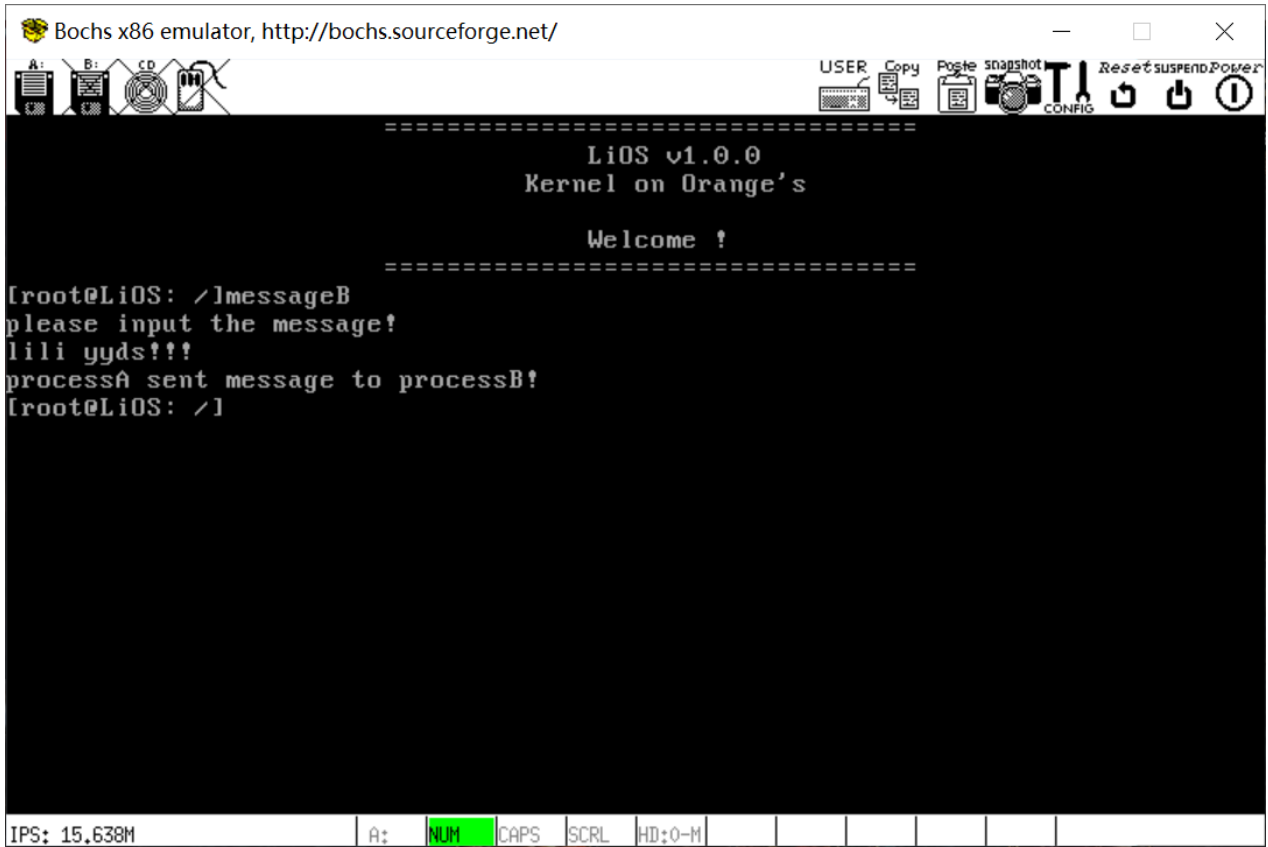
进程间通信

◆ 描述

用户输入“messageA/B/C”向进程A/B/C发送消息，接下来输入想要发送的消息，发送成功则会在两边进程进行提示。（按下alt+f1/f2/f3可切换到进程A/B/C对应的终端）

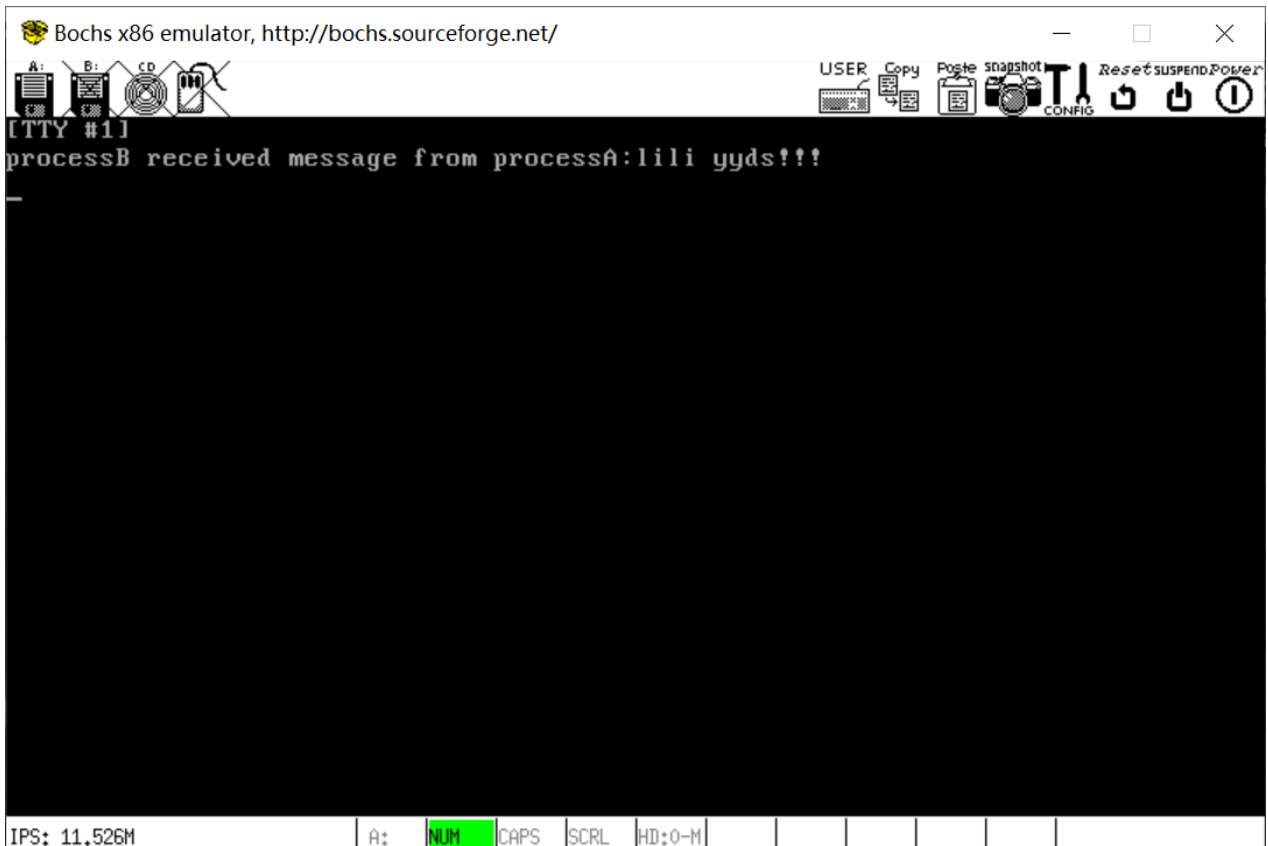
◆ 实现方法

通过调用messageA/B/C()API，利用send_rcv函数，将输入的文本作为消息的一个参数，进行进程间的消息发送与接收来达成进程间通信。



```
Bochs x86 emulator, http://bochs.sourceforge.net/
=====
LiOS v1.0.0
Kernel on Orange's

Welcome !
=====
[root@LiOS: /]messageB
please input the message!
lili yyds!!!
processA sent message to processB!
[root@LiOS: /]
IPS: 15.638M | A: NUM | CAPS | SCRL | HD:0-M | | | | | | | |
```



进程切换

◆ 描述

用户输入“processA/B/C”切换到进程A/B/C。（按下alt+f1/f2/f3可切换到进程A/B/C对应的终端）用户可在切换后的进程进行后续操作，比如向进程A/B/C发送消息。

<<<<<<< HEAD

◆ 实现方法

通过调用processA/B/C()API，利用进程接收消息的阻塞状态，通过进程间通信更改process_running变量确保当前只有一个进程在运行，其他两个进程处于接收消息的阻塞状态，来实现进程的切换。

=====

◆ 实现方法

8337b2543661ddfd353d8af7c646941afbb0d983

通过调用processA/B/C()API，利用进程接收消息的阻塞状态，通过进程间通信更改process_running变量确保当前只有一个进程在运行，其他两个进程处于接收消息的阻塞状态，来实现进程的切换。



成员分工

姓名	学号	分工	占比
陈子杰	1850106	开机动画功能实现；日历、时间功能实现；2048游戏功能实现	20%
马天放	1750773	指令、帮助功能实现；文件夹隐藏、上锁功能实现	20%
黎力	1853549	文件系统功能实现；推箱子小游戏实现	20%
黄伟杰	1850107	扫雷小游戏功能实现；井字棋小游戏功能实现	20%
霍定镭	1850110	进程系统功能实现；进程间通信、进程切换功能实现	20%