OS课程设计文档 for CYTUZ（Software Requirements Specification for CYTUZ)

OS课程设计文档 for CYTUZ（Software Requirements Specification for CYTUZ)

- 1.项目简介
  - 1.1 项目说明
  - 1.2 项目环境
  - 1.3 成员分工
- 2.功能说明
  - 2.1 配置指南
  - 2.2 功能展示
    - 2.2.1 开机界面
    - 2.2.2 帮助界面
    - 2.2.3 文件系统
    - 2.2.4 用户级应用：扫雷
    - 2.2.5 进程管理
    - 2.2.6 game2048
    - 2.2.7 猜数字
    - 2.2.8 五子棋
    - 2.2.9 井字棋
    - 2.2.10 算排列
    - 2.2.11 四则算法计算器
    - 2.2.12 推箱子
- 3.设计与实现
  - 3.1 文件系统
  - 3.2 用户级应用：扫雷
  - 3.3 IO系统-键盘部分
  - 3.4 进程管理部分
  - 3.5 内存管理（子进程生成）部分
  - 3.6 游戏2048
  - 3.7 游戏井字棋
  - 3.8 游戏算排列
  - 3.9 游戏计算器
  - 3.10 游戏猜数字
  - 3.11 游戏双人五子棋
  - 3.12游戏推箱子

1.项目简介

1.1 项目说明

本项目基于ORANGES：一个操作系统的实现一书，完成了一个简单的操作系统CYTUZ，该系统实现了课程设计中要求的三个难度：

B难度：

1）改进了文件系统，详见本文档3.1部分
2）重写了IO系统键盘输入显示部分
3）改进了进程管理部分
4）重写了内存管理部分

D难度：

1）实现了开机动画
1）实现了用户级应用扫雷
2）实现了用户级应用游戏2048
3）实现了用户级应用猜数字
4）实现了用户级应用双人五子棋
5）实现了用户级应用井字棋
6）实现了用户级应用算排列
7）实现了用户级应用四则算法计算器

8）实现了用户级应用推箱子

1.2 项目环境

- 开发环境：ubuntu-20.04 64位系统
- 开发工具：Bochs 2.6.9
- 开发语言：C语言

1.3 成员分工

- 1852141 李德涛：文件系统、框架构建、用户级应用扫雷
- 1853201 侯祖光：IO系统键盘部分重写
- 1852140 上官宇飞：进程管理、用户级应用游戏2048
- 1852715 吴庭辉：开机动画，内存管理（生成子进程），用户级应用猜数字，五子棋（双人），井字棋（AI对战），算排列，四则算法计算器
- 1851910 田原驰： 用户级应用推箱子

2.功能说明

2.1 配置指南

- 本项目默认运行环境为64位Ubuntu，如果是在32 位系统上运行需要修改根目录下Makefile中的LD及gcc参数，删除-m32和-m elf_i386。此外，Bochsrc文件中的romimage，vgaromimage和keymap的路径需要根据本机Bochs的具体安装路径进行修改。
- 解压后在终端中进入项目目录，再从80m.tar.xz中解压出80m.img，终端命令如下:

> unar 80m.tar.xz

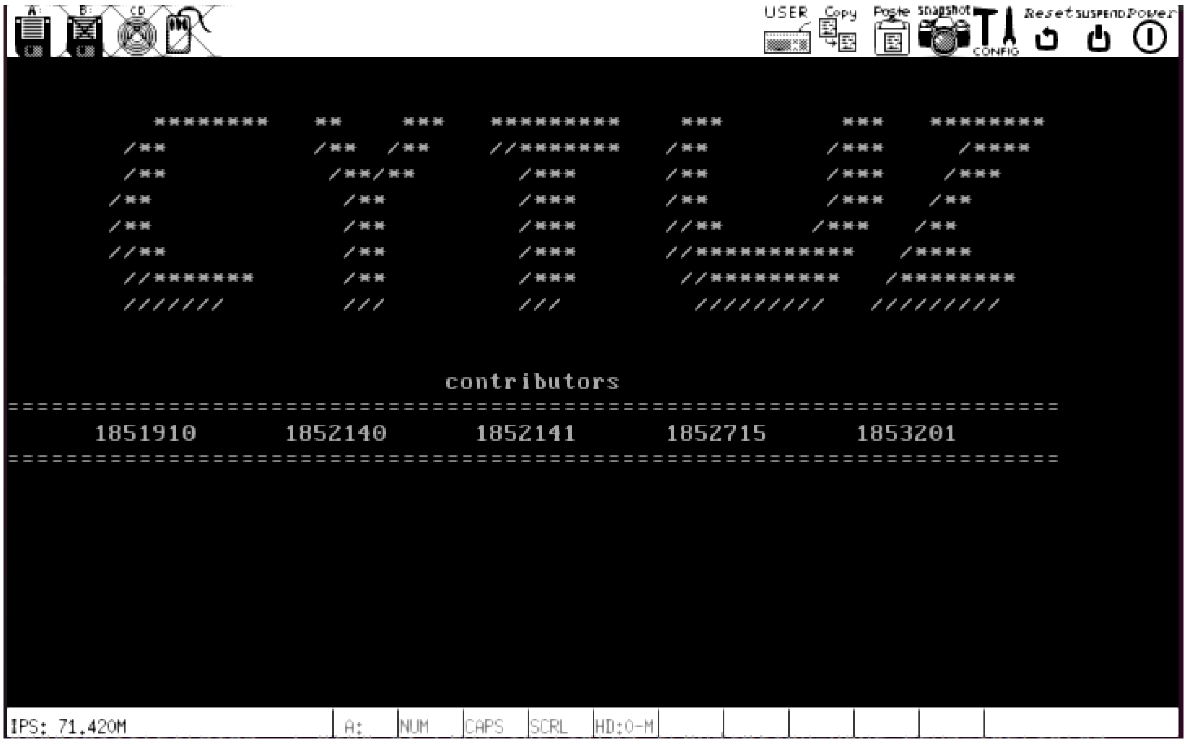
- 编译运行终端命令：

> sudo make image
>
> bochs -f bochsrc

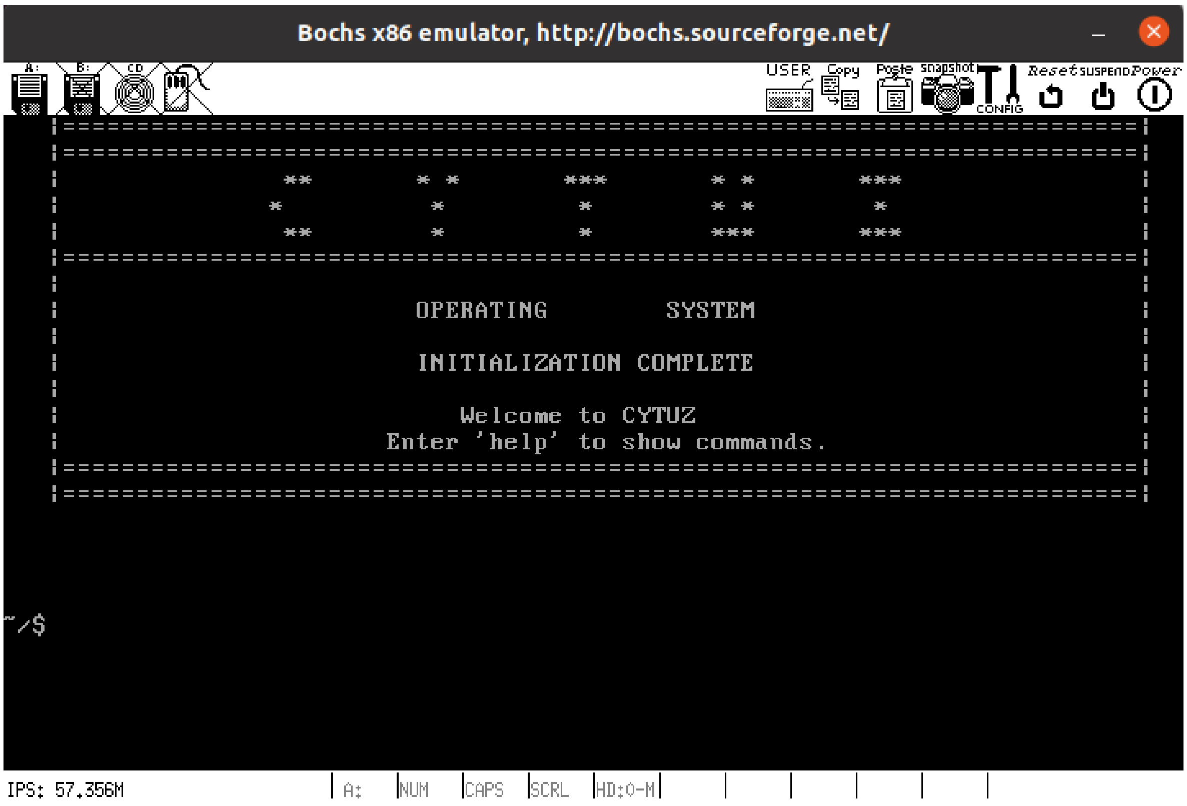
- 运行Bochs后输入6和c退出调试模式即可启动CYTUZ系统。

2.2 功能展示

2.2.1 开机界面



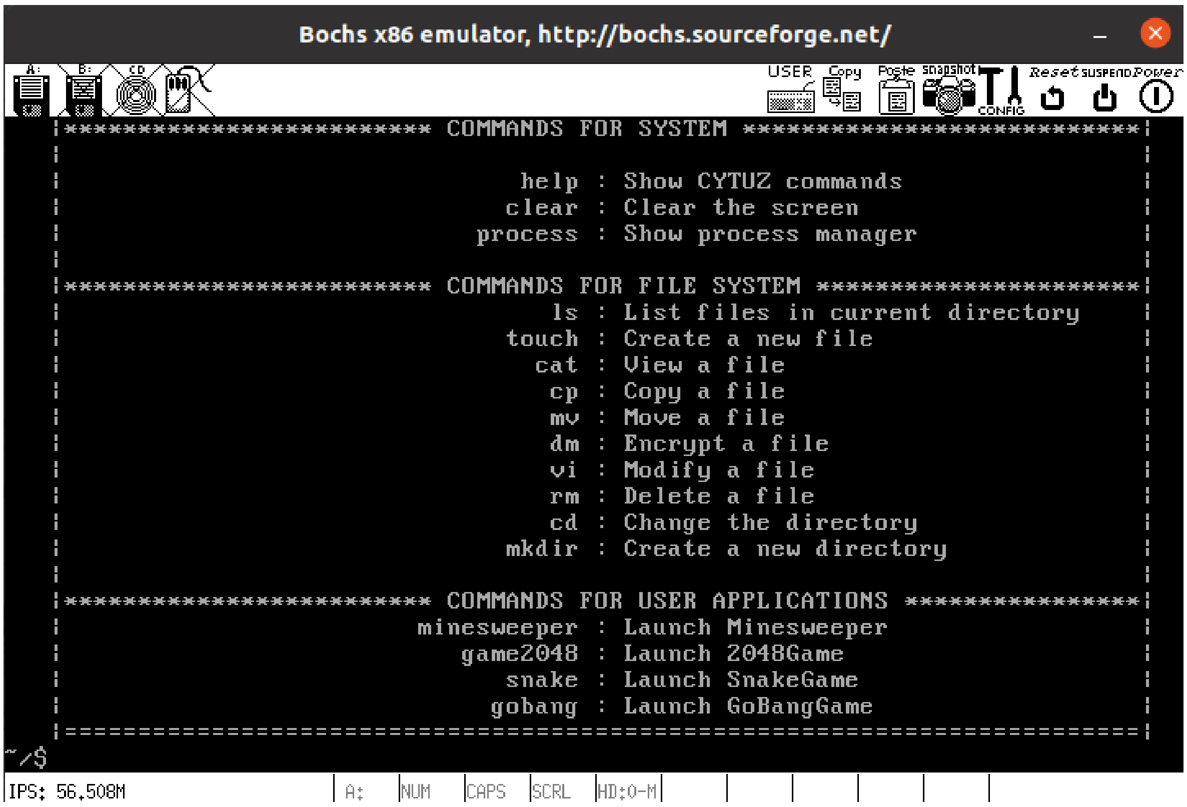
```
          ********   **    ***   *********    ***        ***   ********
        /**          /**  /**    //*******   /**        /***     /****
        /**           /**/**       /***      /**        /***    /***
       /**             /**         /***      /**        /***   /**
       /**             /**         /***      //**      /***   /**
       //**            /**         /***      //***********   /****
        //*******      /**         /***       //**********  /********
        ///////        ///         ///         /////////   /////////


                              contributors
========================================================================
      1851910      1852140      1852141      1852715      1853201
========================================================================
```



```
   |=========================================================================|
   |=========================================================================|
   |               **       * *       ***       * *       ***                |
   |              *          *         *        * *        *                 |
   |               **        *         *        ***       ***                |
   |=========================================================================|
   |                                                                         |
   |                        OPERATING        SYSTEM                          |
   |                                                                         |
   |                        INITIALIZATION COMPLETE                          |
   |                                                                         |
   |                           Welcome to CYTUZ                              |
   |                      Enter 'help' to show commands.                     |
   |=========================================================================|
   |=========================================================================|




~/$
```

2.2.2 帮助界面

提供CYTUZ系统的全部指令及其详细描述，输入命令即可实现相应操作。

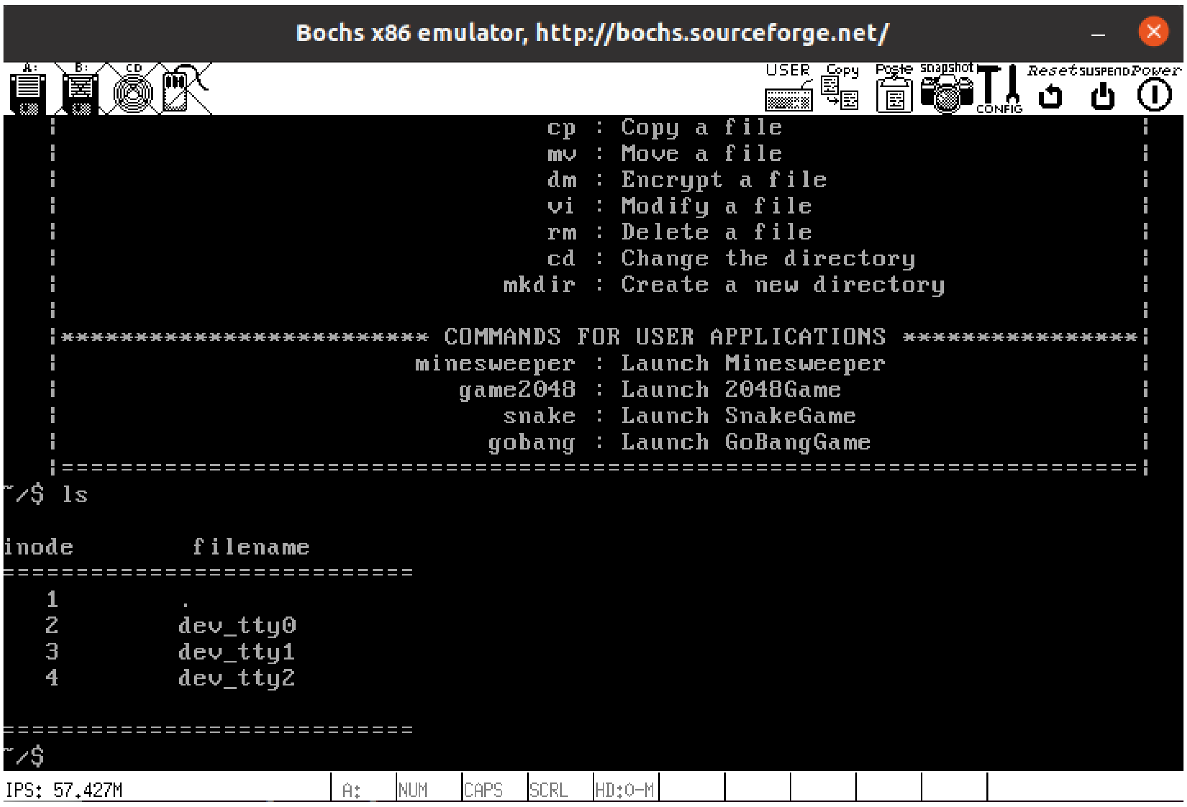


```
|************************** COMMANDS FOR SYSTEM *****************************|
|                                                                            |
|                                help : Show CYTUZ commands                  |
|                               clear : Clear the screen                     |
|                             process : Show process manager                 |
|                                                                            |
|************************** COMMANDS FOR FILE SYSTEM ************************|
|                                  ls : List files in current directory      |
|                               touch : Create a new file                    |
|                                 cat : View a file                          |
|                                  cp : Copy a file                          |
|                                  mv : Move a file                          |
|                                  dm : Encrypt a file                       |
|                                  vi : Modify a file                        |
|                                  rm : Delete a file                        |
|                                  cd : Change the directory                 |
|                               mkdir : Create a new directory               |
|                                                                            |
|************************** COMMANDS FOR USER APPLICATIONS ******************|
|                         minesweeper : Launch Minesweeper                   |
|                            game2048 : Launch 2048Game                      |
|                               snake : Launch SnakeGame                     |
|                              gobang : Launch GoBangGame                    |
|============================================================================|
~/$
IPS: 56.508M   A:  NUM  CAPS  SCRL  HD:0-M
```

2.2.3 文件系统

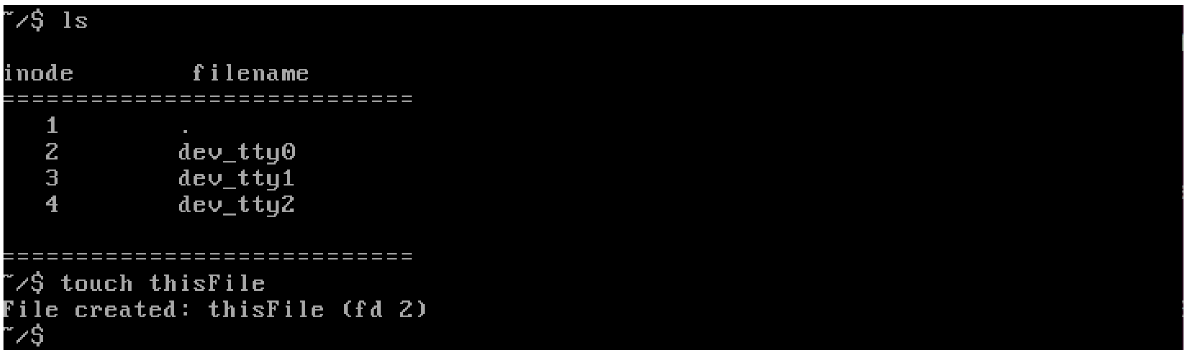
CYTUZ文件系统命令行指令模仿了Linux的命名习惯，总共提供了9种不同的操作，改进了OrangeS的文件系统，使之可以创建多级目录。

ls：显示当前工作目录下的文件



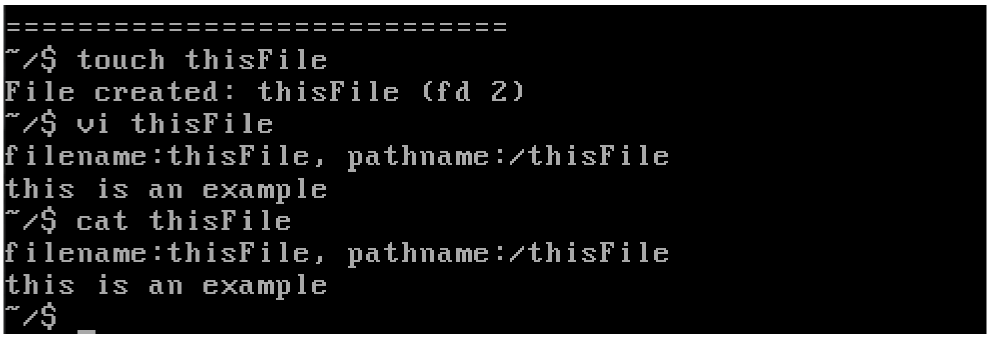
```
|                                  cp : Copy a file                          |
|                                  mv : Move a file                          |
|                                  dm : Encrypt a file                       |
|                                  vi : Modify a file                        |
|                                  rm : Delete a file                        |
|                                  cd : Change the directory                 |
|                               mkdir : Create a new directory               |
|                                                                            |
|************************** COMMANDS FOR USER APPLICATIONS ******************|
|                         minesweeper : Launch Minesweeper                   |
|                            game2048 : Launch 2048Game                      |
|                               snake : Launch SnakeGame                     |
|                              gobang : Launch GoBangGame                    |
|============================================================================|
~/$ ls

inode        filename
============================
   1           .
   2           dev_tty0
   3           dev_tty1
   4           dev_tty2

============================
~/$
IPS: 57.427M   A:  NUM  CAPS  SCRL  HD:0-M
```

touch：创建新的文件



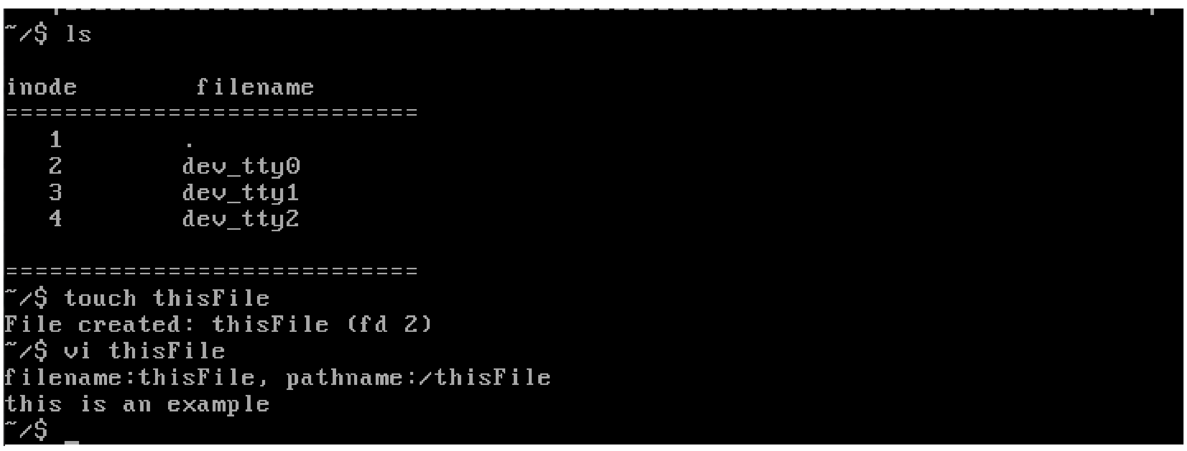
```
~/$ ls

inode        filename
============================
   1         .
   2        dev_tty0
   3        dev_tty1
   4        dev_tty2

============================
~/$ touch thisFile
File created: thisFile (fd 2)
~/$
```

cat：查看文件



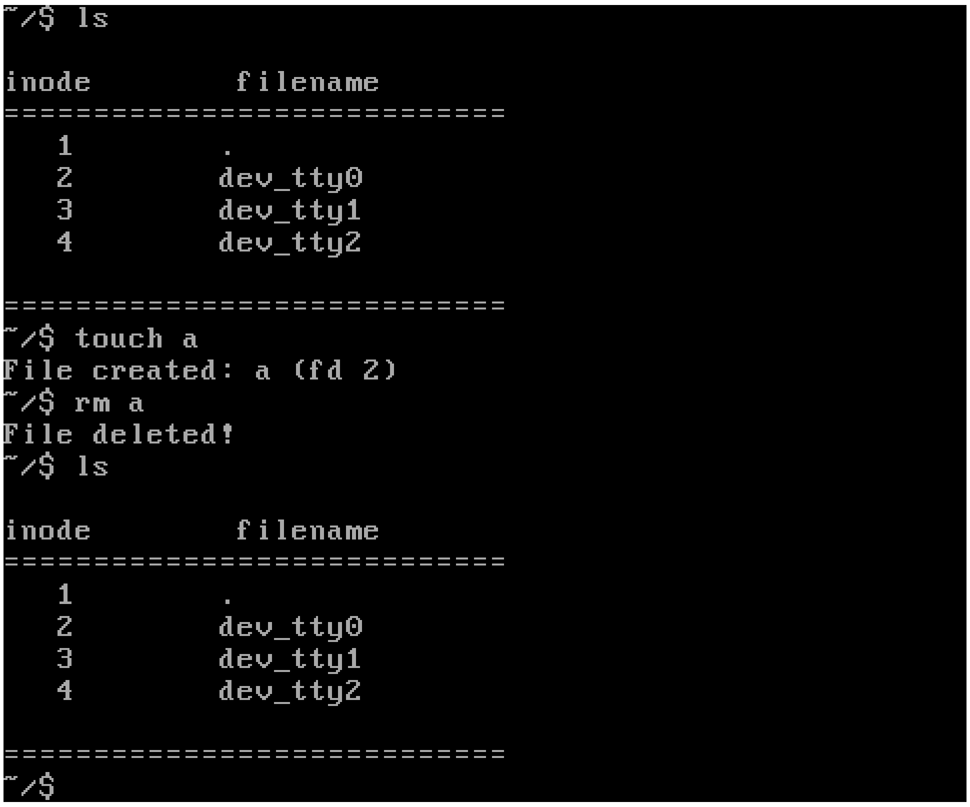
```
============================
~/$ touch thisFile
File created: thisFile (fd 2)
~/$ vi thisFile
filename:thisFile, pathname:/thisFile
this is an example
~/$ cat thisFile
filename:thisFile, pathname:/thisFile
this is an example
~/$ _
```

vi：修改文件



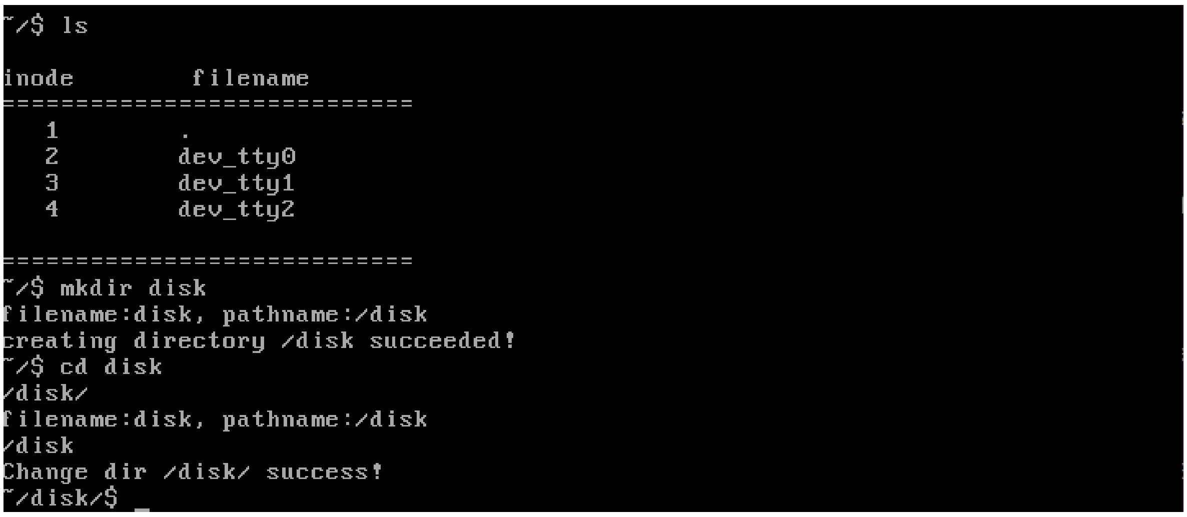
```
~/$ ls

inode        filename
============================
   1         .
   2        dev_tty0
   3        dev_tty1
   4        dev_tty2

============================
~/$ touch thisFile
File created: thisFile (fd 2)
~/$ vi thisFile
filename:thisFile, pathname:/thisFile
this is an example
~/$ _
```

rm：删除文件



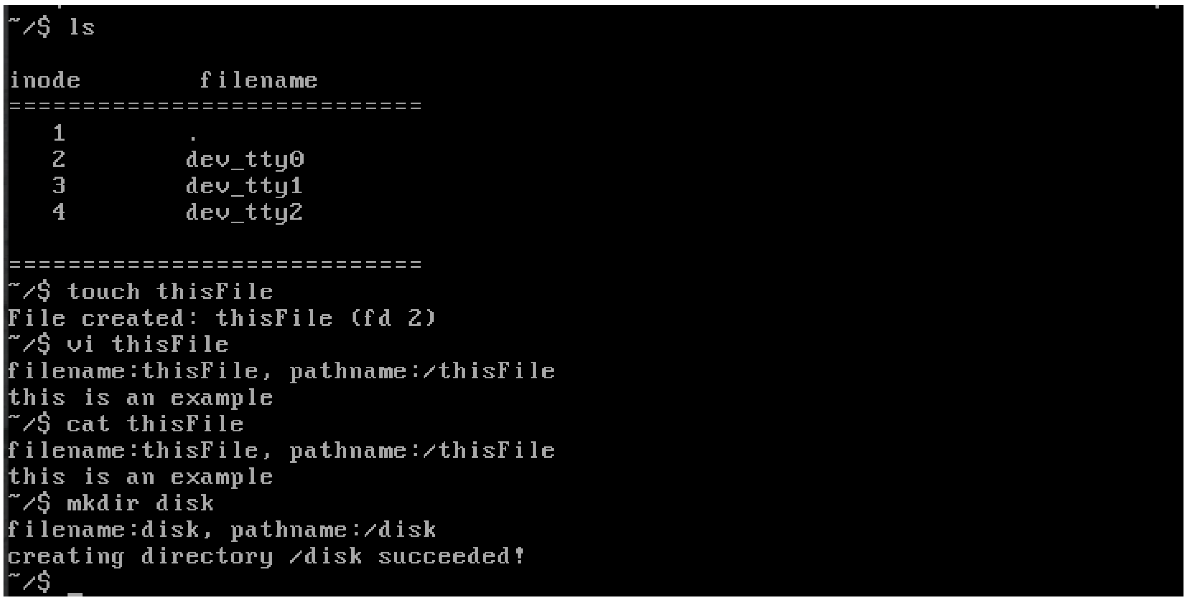
```
~/$ ls

inode         filename
============================
   1          .
   2         dev_tty0
   3         dev_tty1
   4         dev_tty2

============================
~/$ touch a
File created: a (fd 2)
~/$ rm a
File deleted!
~/$ ls

inode         filename
============================
   1          .
   2         dev_tty0
   3         dev_tty1
   4         dev_tty2

============================
~/$
```

cd：改变当前工作目录



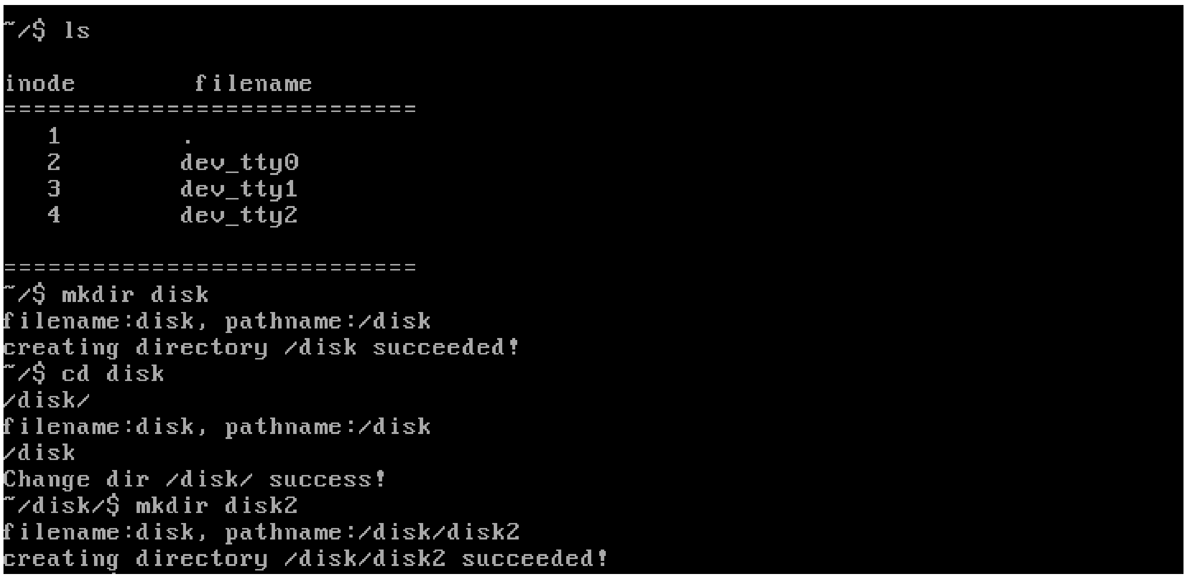
```
~/$ ls

inode         filename
============================
   1          .
   2         dev_tty0
   3         dev_tty1
   4         dev_tty2

============================
~/$ mkdir disk
filename:disk, pathname:/disk
creating directory /disk succeeded!
~/$ cd disk
/disk/
filename:disk, pathname:/disk
/disk
Change dir /disk/ success!
~/disk/$ _
```

mkdir：创建新的路径



```
~/$ ls

inode         filename
============================
   1          .
   2         dev_tty0
   3         dev_tty1
   4         dev_tty2

============================
~/$ touch thisFile
File created: thisFile (fd 2)
~/$ vi thisFile
filename:thisFile, pathname:/thisFile
this is an example
~/$ cat thisFile
filename:thisFile, pathname:/thisFile
this is an example
~/$ mkdir disk
filename:disk, pathname:/disk
creating directory /disk succeeded!
~/$ _
```

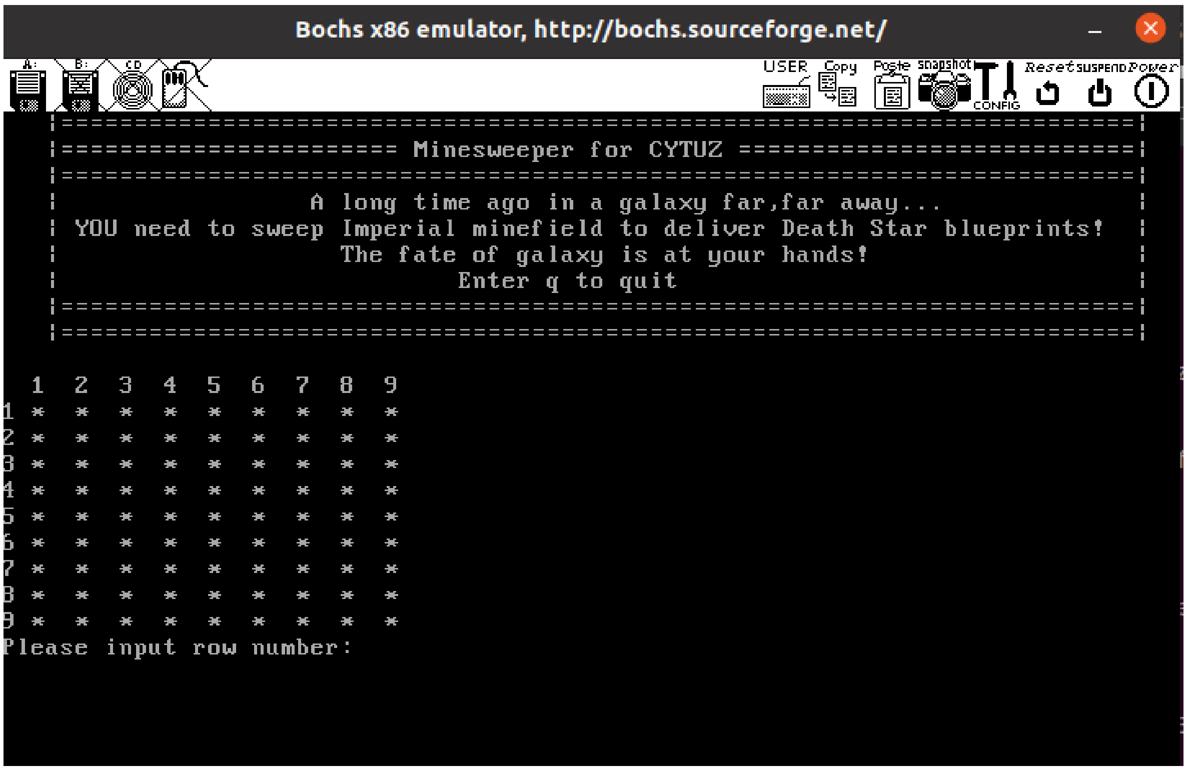
二级目录：



```
~/$ ls

inode        filename
============================
   1          .
   2          dev_tty0
   3          dev_tty1
   4          dev_tty2

============================
~/$ mkdir disk
filename:disk, pathname:/disk
creating directory /disk succeeded!
~/$ cd disk
/disk/
filename:disk, pathname:/disk
/disk
Change dir /disk/ success!
~/disk/$ mkdir disk2
filename:disk, pathname:/disk/disk2
creating directory /disk/disk2 succeeded!
```

2.2.4 用户级应用：扫雷

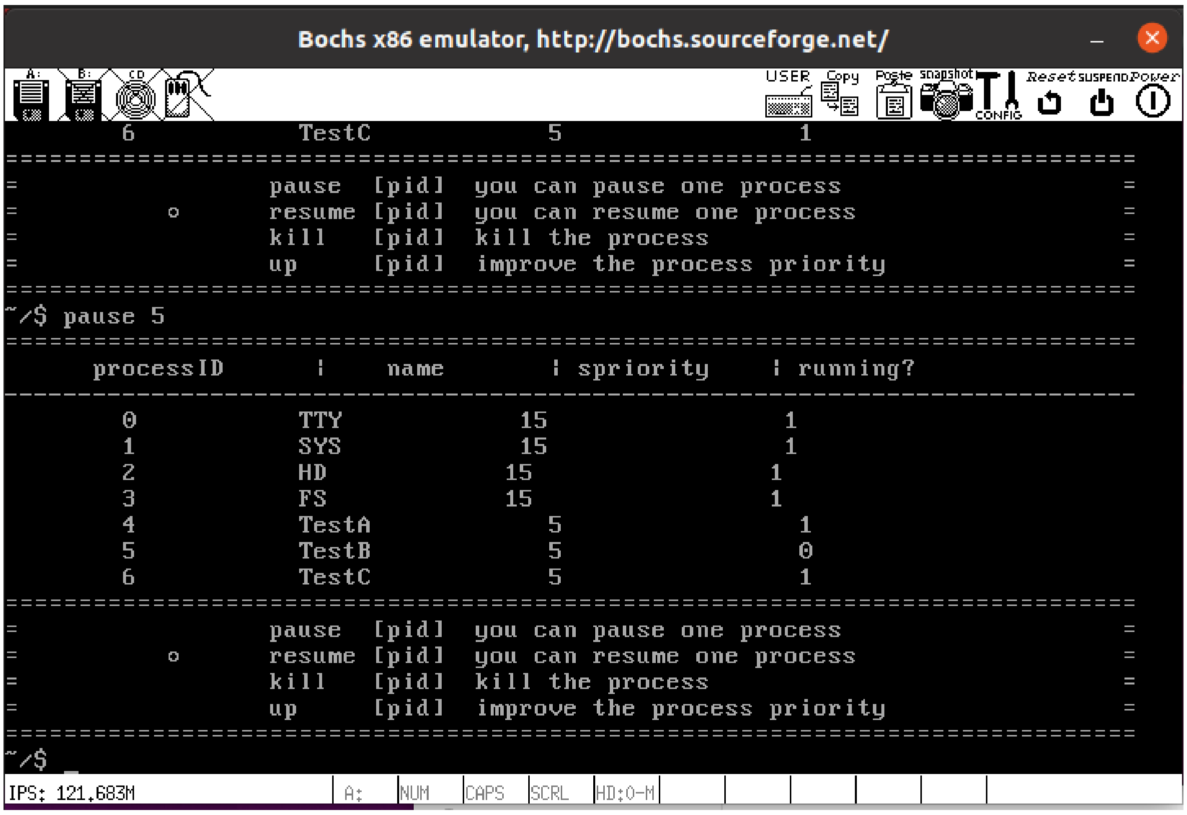
该功能实现了一个扫雷小游戏，可以在一个10*10的雷区中随机生成10颗雷供玩家游玩。



```
|============================================================================|
|====================== Minesweeper for CYTUZ ==========================|
|============================================================================|
|                    A long time ago in a galaxy far,far away...             |
| YOU need to sweep Imperial minefield to deliver Death Star blueprints!     |
|                    The fate of galaxy is at your hands!                    |
|                              Enter q to quit                              |
|============================================================================|
|============================================================================|

  1  2  3  4  5  6  7  8  9
1 *  *  *  *  *  *  *  *  *
2 *  *  *  *  *  *  *  *  *
3 *  *  *  *  *  *  *  *  *
4 *  *  *  *  *  *  *  *  *
5 *  *  *  *  *  *  *  *  *
6 *  *  *  *  *  *  *  *  *
7 *  *  *  *  *  *  *  *  *
8 *  *  *  *  *  *  *  *  *
9 *  *  *  *  *  *  *  *  *
Please input row number:
```

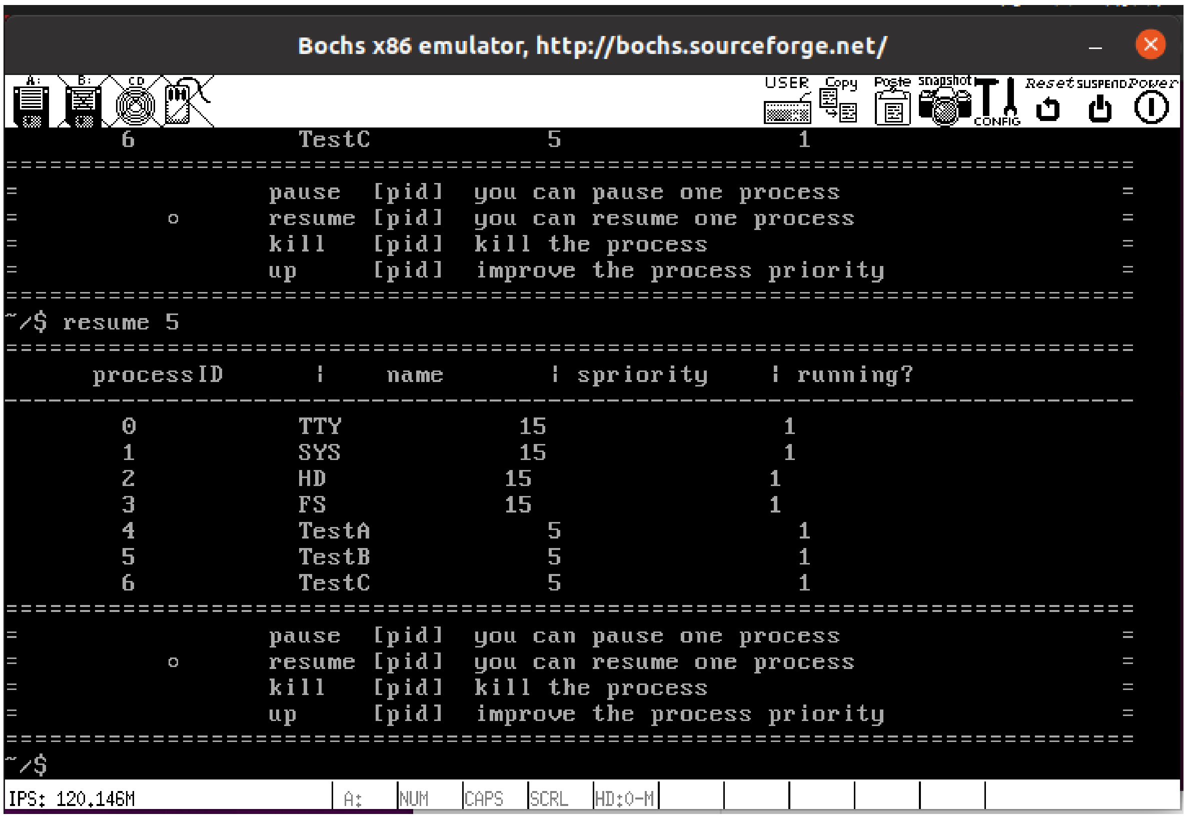
2.2.5 进程管理

pause：暂停一个进程



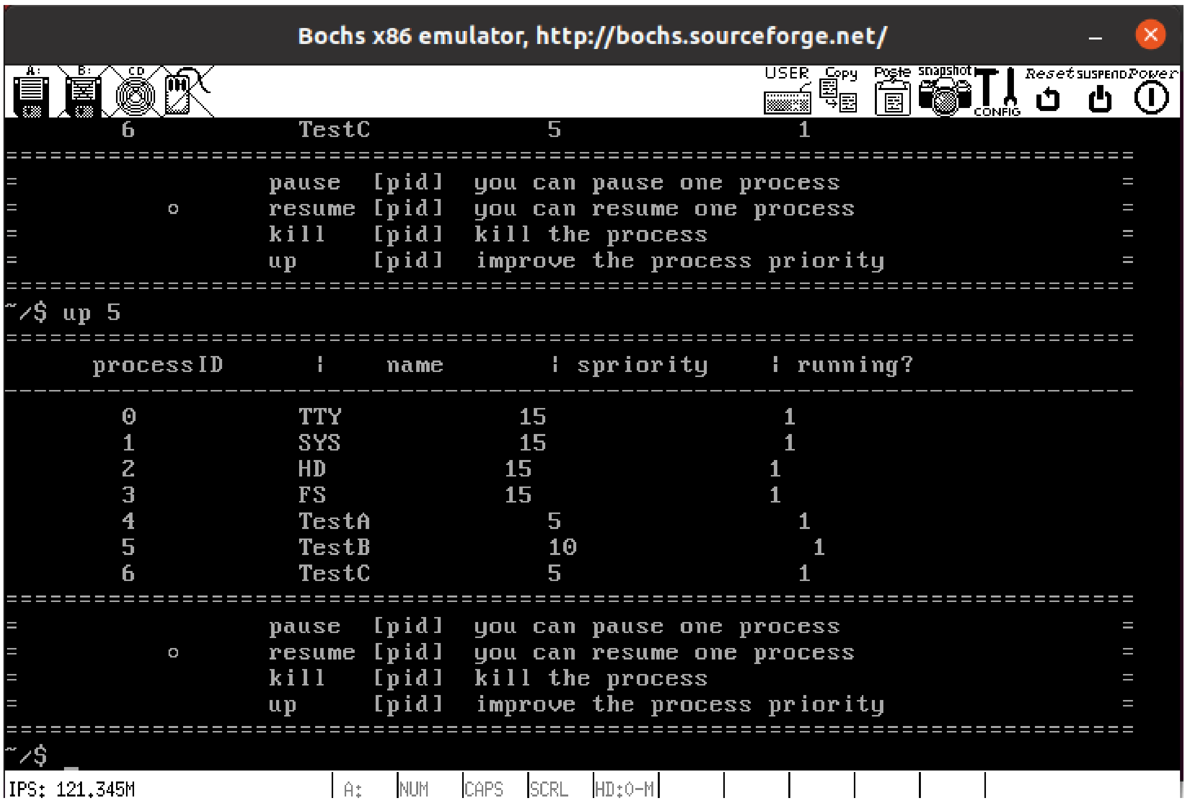
```
        6            TestC            5                 1
=============================================================================
=                 pause  [pid]  you can pause one process                   =
=          o      resume [pid]  you can resume one process                  =
=                 kill   [pid]  kill the process                            =
=                 up     [pid]  improve the process priority                =
=============================================================================
~/$ pause 5
=============================================================================
      processID       ¦     name        ¦ spriority     ¦ running?
-----------------------------------------------------------------------------
        0            TTY               15                1
        1            SYS               15                1
        2            HD               15                1
        3            FS               15                1
        4            TestA            5                 1
        5            TestB            5                 0
        6            TestC            5                 1
=============================================================================
=                 pause  [pid]  you can pause one process                   =
=          o      resume [pid]  you can resume one process                  =
=                 kill   [pid]  kill the process                            =
=                 up     [pid]  improve the process priority                =
=============================================================================
~/$ _
```

resume：恢复一个进程



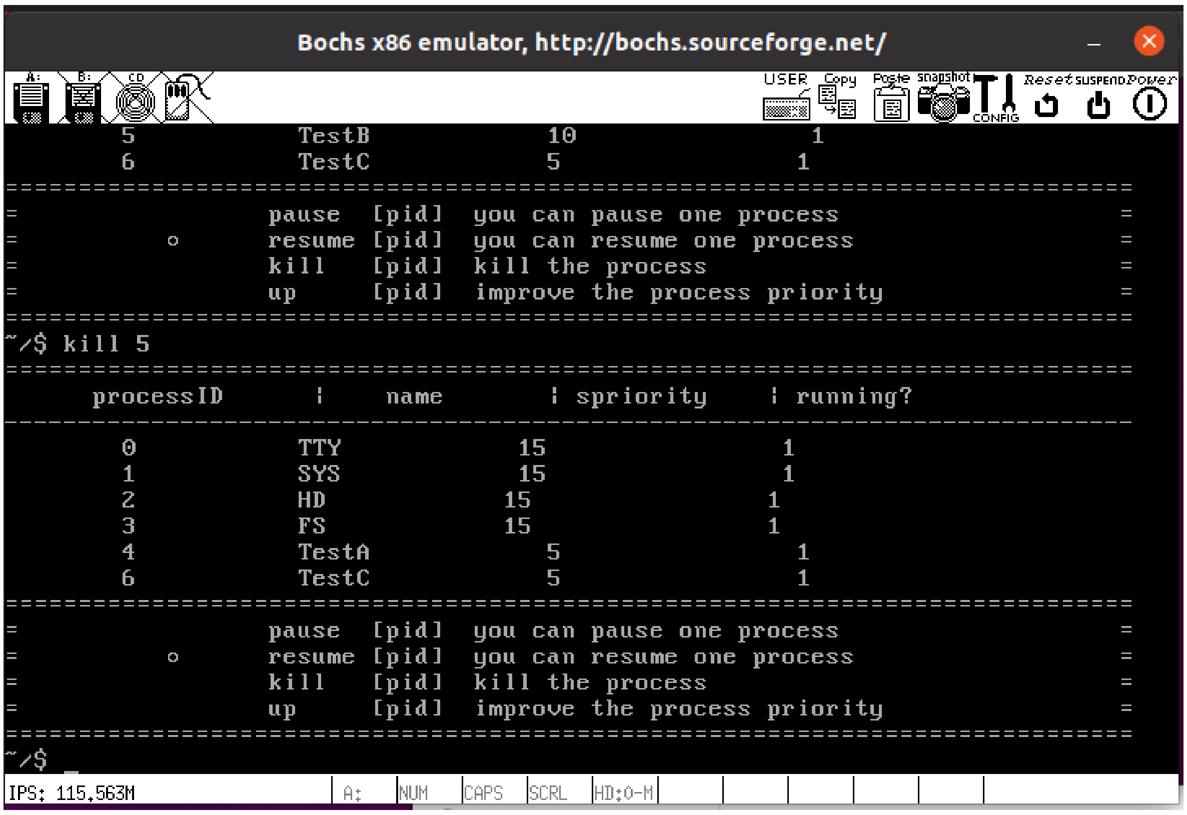
```
        6            TestC            5                 1
=============================================================================
=                 pause  [pid]  you can pause one process                   =
=          o      resume [pid]  you can resume one process                  =
=                 kill   [pid]  kill the process                            =
=                 up     [pid]  improve the process priority                =
=============================================================================
~/$ resume 5
=============================================================================
      processID       ¦     name        ¦ spriority     ¦ running?
-----------------------------------------------------------------------------
        0            TTY               15                1
        1            SYS               15                1
        2            HD               15                1
        3            FS               15                1
        4            TestA            5                 1
        5            TestB            5                 1
        6            TestC            5                 1
=============================================================================
=                 pause  [pid]  you can pause one process                   =
=          o      resume [pid]  you can resume one process                  =
=                 kill   [pid]  kill the process                            =
=                 up     [pid]  improve the process priority                =
=============================================================================
~/$
```

up：使某进程优先度乘二



```
        6           TestC            5                1
=============================================================================
=                  pause  [pid]  you can pause one process                  =
=          o       resume [pid]  you can resume one process                 =
=                  kill   [pid]  kill the process                           =
=                  up     [pid]  improve the process priority               =
=============================================================================
~/$ up 5
=============================================================================
      processID     |     name      | spriority    | running?
-----------------------------------------------------------------------------
        0           TTY             15               1
        1           SYS             15               1
        2           HD             15              1
        3           FS             15              1
        4           TestA            5                1
        5           TestB             10               1
        6           TestC            5                1
=============================================================================
=                  pause  [pid]  you can pause one process                  =
=          o       resume [pid]  you can resume one process                 =
=                  kill   [pid]  kill the process                           =
=                  up     [pid]  improve the process priority               =
=============================================================================
~/$ _
IPS: 121.345M            A:   NUM   CAPS   SCRL   HD:0-M
```

kill：杀死某进程（将其从进程表中去除）



```
        5           TestB             10               1
        6           TestC            5                1
=============================================================================
=                  pause  [pid]  you can pause one process                  =
=          o       resume [pid]  you can resume one process                 =
=                  kill   [pid]  kill the process                           =
=                  up     [pid]  improve the process priority               =
=============================================================================
~/$ kill 5
=============================================================================
      processID     |     name      | spriority    | running?
-----------------------------------------------------------------------------
        0           TTY             15               1
        1           SYS             15               1
        2           HD             15              1
        3           FS             15              1
        4           TestA            5                1
        6           TestC            5                1
=============================================================================
=                  pause  [pid]  you can pause one process                  =
=          o       resume [pid]  you can resume one process                 =
=                  kill   [pid]  kill the process                           =
=                  up     [pid]  improve the process priority               =
=============================================================================
~/$ _
IPS: 115.563M            A:   NUM   CAPS   SCRL   HD:0-M
```

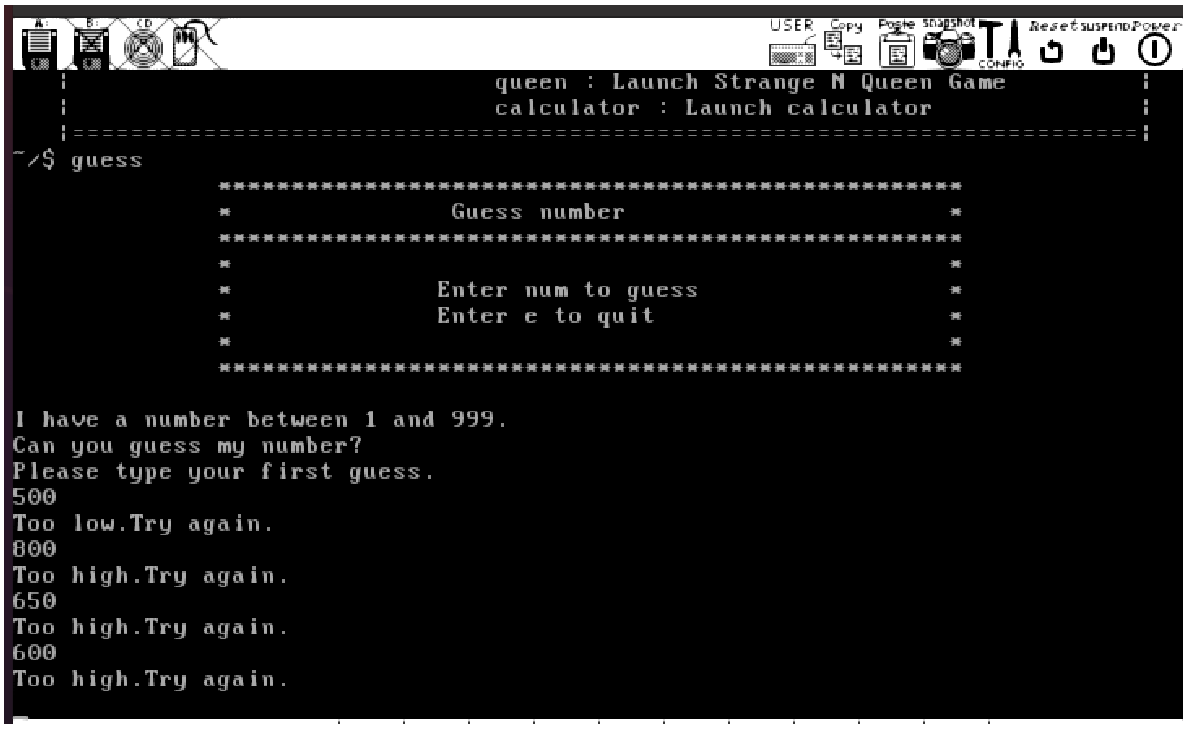
2.2.6 game2048

该功能实现了2048小游戏，在4*4方格内进行游戏

image-20200816224647568

2.2.7 猜数字

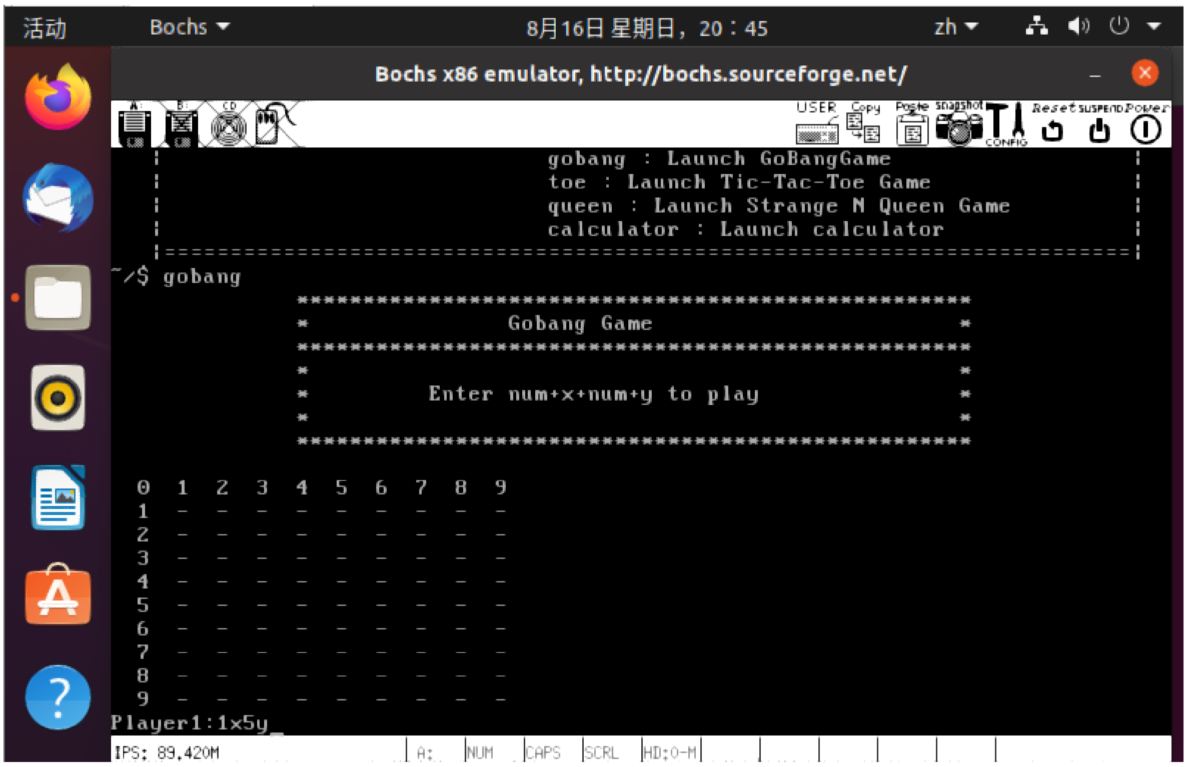
该功能实现了猜数字小游戏，随机生成数字，进行猜测，给出高低提示，按e退出游戏



```
        |                          queen : Launch Strange N Queen Game                      |
        |                          calculator : Launch calculator                           |
        |===================================================================================|
~/$ guess
              ****************************************************
              *                Guess number                      *
              ****************************************************
              *                                                  *
              *              Enter num to guess                  *
              *              Enter e to quit                     *
              *                                                  *
              ****************************************************

I have a number between 1 and 999.
Can you guess my number?
Please type your first guess.
500
Too low.Try again.
800
Too high.Try again.
650
Too high.Try again.
600
Too high.Try again.
```

2.2.8 五子棋

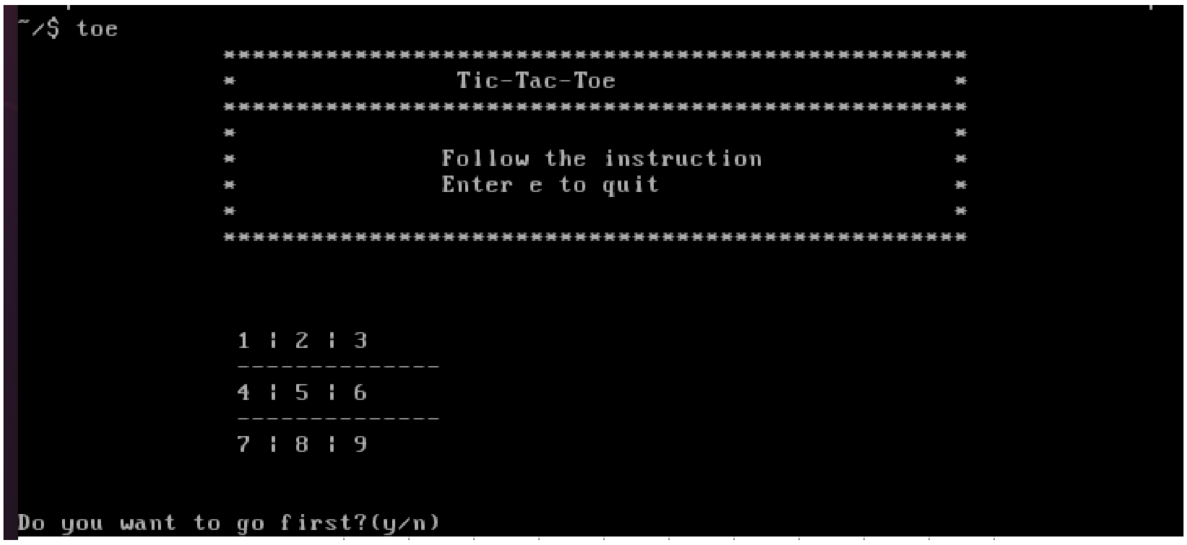
该功能实现了五子棋小游戏，在棋盘内进行双人对战



```
        |                          gobang : Launch GoBangGame                     |
        |                          toe : Launch Tic-Tac-Toe Game                  |
        |                          queen : Launch Strange N Queen Game            |
        |                          calculator : Launch calculator                 |
        |=========================================================================|
~/$ gobang
            ****************************************************
            *                Gobang Game                       *
            ****************************************************
            *                                                  *
            *         Enter num+x+num+y to play                *
            *                                                  *
            ****************************************************

  0  1  2  3  4  5  6  7  8  9
  1  -  -  -  -  -  -  -  -  -
  2  -  -  -  -  -  -  -  -  -
  3  -  -  -  -  -  -  -  -  -
  4  -  -  -  -  -  -  -  -  -
  5  -  -  -  -  -  -  -  -  -
  6  -  -  -  -  -  -  -  -  -
  7  -  -  -  -  -  -  -  -  -
  8  -  -  -  -  -  -  -  -  -
  9  -  -  -  -  -  -  -  -  -
Player1:1x5y_
```

2.2.9 井字棋

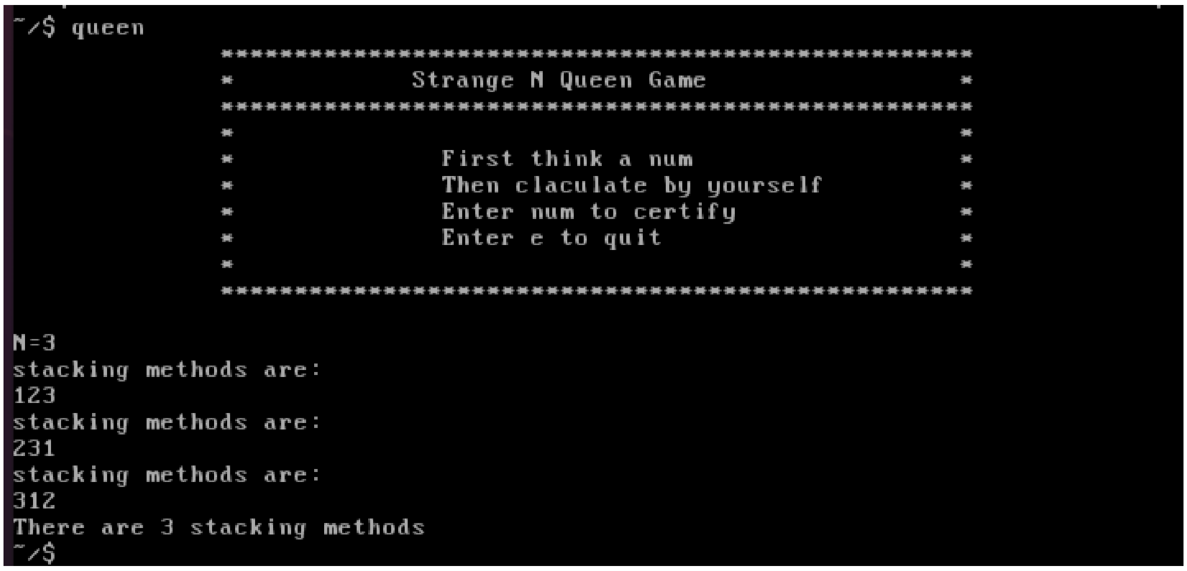
该功能实现了井字棋小游戏，在棋盘内与AI进行对战游戏



```
~/$ toe
                ************************************************
                *              Tic-Tac-Toe                     *
                ************************************************
                *                                              *
                *             Follow the instruction           *
                *             Enter e to quit                  *
                *                                              *
                ************************************************



                 1 | 2 | 3
                 -------------
                 4 | 5 | 6
                 -------------
                 7 | 8 | 9

Do you want to go first?(y/n)
```

2.2.10 算排列

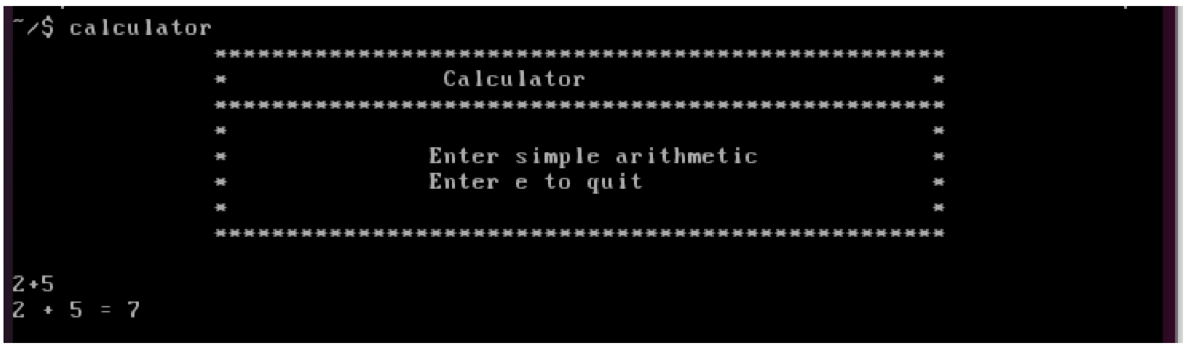
该功能实现了算排列小游戏，输入长宽，然后默算排列种类，最后看是否与计算结果一致



```
~/$ queen
                ************************************************
                *           Strange N Queen Game               *
                ************************************************
                *                                              *
                *             First think a num                *
                *             Then claculate by yourself       *
                *             Enter num to certify             *
                *             Enter e to quit                  *
                *                                              *
                ************************************************

N=3
stacking methods are:
123
stacking methods are:
231
stacking methods are:
312
There are 3 stacking methods
~/$
```

2.2.11 四则算法计算器

该功能实现了输入简单的四则算法，得出答案



```
~/$ calculator
                ************************************************
                *              Calculator                      *
                ************************************************
                *                                              *
                *             Enter simple arithmetic          *
                *             Enter e to quit                  *
                *                                              *
                ************************************************

2+5
2 + 5 = 7
```

2.2.12 推箱子

该功能实现了推箱子小游戏，在一个空间中，把木箱放到指定的位置，出现箱子无法移动或者通道被堵住的情况即为失败，推到指定位置则游戏成功

3.设计与实现

3.1 文件系统

- ls： shell调用ls(current_dirr)，向文件系统发送系统命令，进而执行do_ls()，遍历current_dirr下的子节点，打印该目录下的所有文件/文件夹。

```
for (i = 0; i < nr_dir_blks; i++)
   {
       RD_SECT(dir_inode->i_dev, dir_blk0_nr + i);

       pde = (struct dir_entry *)fsbuf;

       for (j = 0; j < SECTOR_SIZE / DIR_ENTRY_SIZE; j++, pde++)
       {
           /*struct inode *n = find_inode(pde->inode_nr);*/
           printl("  %2d        %s\n", pde->inode_nr , pde->name);
           if (++m >= nr_dir_entries){
               printl("\n");
               break;
           }
       }
       if (m > nr_dir_entries) //[> all entries have been iterated <]
           break;
   }
```

- touch：调用open()创建新的文件，若创建失败则提示检查文件名。否则调用write()完成写入。

```
           fd = open(arg1, O_CREAT | O_RDWR);
           if (fd == -1)
           {
               printf("Failed to create file! Please check the filename!\n");
               continue ;
           }
           write(fd, buf, 1);
           printf("File created: %s (fd %d)\n", arg1, fd);
           close(fd);
```

- cat：调用open()查看文件，若失败则提示检查文件名，再调用read()访问文件。

```
           fd = open(arg1, O_RDWR);
           if (fd == -1)
           {
               printf("Failed to open file! Please check the filename!\n");
               continue ;
           }
           read(fd, buf, 1024);
           close(fd);
```

（后面部分对 文件是否存在 的检查代码不再赘述）

- cp：用户输入源地址和先获取文件内容，若文件存在则使用temp存放文件内容，再调用write()将temp写入目标地址。

```
            fd = open(arg1, O_RDWR);
            int tail = read(fd_stdin, rdbuf, 512);
            rdbuf[tail] = 0;

            write(fd, rdbuf, tail+1);
            close(fd);
```

- mv：与cp操作类似，但在最后一步加上 unlink(arg1)这一函数将源文件删除。

```
unlink(arg1);
```

- vi：系统先使用open()找到目标文件路径，然后调用write()对文件内容进行更改，将输入的内容写入目标文件。

```
            fd = open(arg1, O_RDWR);
            if (fd == -1)
            {
                printf("Failed to open file! Please check the filename!\n");
                continue ;
            }
            int tail = read(fd_stdin, rdbuf, 512);
            rdbuf[tail] = 0;

            write(fd, rdbuf, tail+1);
            close(fd);
```

- rm：文件系统调用do_unlink()，递归删除文件夹中的子节点。

```
            int result;
            result = unlink(arg1);
            if (result == 0)
            {
                printf("File deleted!\n");
                continue;
            }
            else
            {
                printf("Failed to delete file! Please check the filename!\n");
                continue;
            }
```

- cd：首先根据输入的路径名判断是进入下一级目录还是返回上一级目录，并根据相应情况修改path。

```
int flag = 0;
    char newPath[512] = {0};
    if (file[0] == '.' && file[1] == '.')
    {
        flag = 1;
        int pos_path = 0;
        int pos_new = 0;
        int i = 0;
        char temp[128] = {0};
        while (path[pos_path] != 0)
        {
            if (path[pos_path] == '/')
            {
                pos_path++;
                if (path[pos_path] == 0)
                    break;
                else
                {
                    temp[i] = '/';
                    temp[i + 1] = 0;
                    i = 0;
                    while (temp[i] != 0)
                    {
                        newPath[pos_new] = temp[i];
                        temp[i] = 0;
                        pos_new++;
                        i++;
                    }
                    i = 0;
                }
            }
            else
            {
                temp[i] = path[pos_path];
                i++;
                pos_path++;
            }
        }
    }
    char absoPath[512];
    char temp[512];
    int pos = 0;
    while (file[pos] != 0)
    {
        temp[pos] = file[pos];
        pos++;
    }
    temp[pos] = '/';
    temp[pos + 1] = 0;
    if (flag == 1)
    {
        temp[0] = 0;
        convert_to_absolute(absoPath, newPath, temp);
    }
    else
        convert_to_absolute(absoPath, path, temp);
    int fd = open(absoPath, O_RDWR);
    if (fd == -1)
        printf("%s is not a directory!\n", absoPath);
    else
        memcpy(path, absoPath, 512);
```

- mkdir：先调用createDir()创建新的路径，再通过fs中修改过的do_mkdir()创建文件夹

```
    int fd = open(absoPath, O_RDWR);

    if (fd != -1)
    {
        printf("Failed to create a new directory with name %s\n", file);
        return;
    }
    mkdir(absoPath);
```

3.2 用户级应用：扫雷

- 该小游戏使用了rand()函数来生成随机的地雷，并使用以下四个函数作为游戏的主体：

-
  ```
      m_init();
      m_set_mine();
      m_display(show);
      m_sweep();
  ```

3.3 IO系统-键盘部分

- 键盘键入时press和release都会产生相应的编码，考虑到组合键以及针对不同情况press和release的分析，仅采用硬件提供的缓冲区不足以解决问题，因此需要一个自定义缓冲区存储键盘键入信息

```
struct kbInbuf {
    // 缓冲区头
    char* head;
    // 缓冲区尾
    char* tail;
    // 字节数
    int count;
    // 缓冲区
    char    buf[KB_IN_BYTES];
};
```

- 缓冲区有编码时，用此函数将编码一个个拿出来

```
PRIVATE u8 getByteFromKbBuffer()
{
    u8 code;

    // 等待输入
    while(kbIn.count <= 0);

    disable_int();
    code = *(kbIn.tail);
    kbIn.tail++;
    if(kbIn.tail == kbIn.buf + KB_IN_BYTES){
        kbIn.tail = kbIn.buf;
    }
    kbIn.count--;
    enable_int();

    return code;
```

```
}
```

- 拿出的编码根据不同的情况进行分析，其中0xE1与0xE0开头的press产生的编码都是多字节的，因此需要单独处理，其他的编码根据组合键以及对应情况进行判断处理
- 例：大写字母需要根据左右shift是否按下以及CapsLock是否亮起来判断输出，对应的状态按键设置全局变量来监控，由缓冲区中读出的编码与预设编码表进行匹配来改变状态

```
PUBLIC void kbRead(TTY* tty)
{
    // 扫描码
    u8 code;

    // 1: make
    // 0: break
    int make;

    // 循环处理用的键位记录
    u32 key;

    u32* keyRow;

    while(kbIn.count > 0){
        codeE0 = 0;
        code = getByteFromKbBuffer();

        // 0xE1和0xE0单独处理，双字节编码
        if(code == 0xE0){
            codeE0 = 1;
            code = getByteFromKbBuffer();
            // printScreen键 pressed
            if(code == 0x2A){
                codeE0 = 0;
                if((code = getByteFromKbBuffer()) == 0xE0){
                    codeE0 = 1;
                    if((code = getByteFromKbBuffer()) == 0x37){
                        key = PRINTSCREEN;
                        make = 1;
                    }
                }
            }
            // release
            else if(code == 0xB7){
                codeE0 = 0;
                if((code = getByteFromKbBuffer()) == 0xE0){
                    codeE0 = 1;
                    if((code = getByteFromKbBuffer()) == 0xAA){
                        key = PRINTSCREEN;
                        make = 0;
                    }
                }
            }
        }
        else if(code == 0xE1){
            u8 pauseCodeArr[] = {0xE1, 0x1D, 0x45, 0xE1, 0x9D, 0xC5};
            int isPause = 1;
            for(int i = 0; i < 6; ++i){
                if(getByteFromKbBuffer() != pauseCodeArr[i]){
```

```
                isPause = 0;
                break;
            }
        }
        if(isPause){
            key = PAUSEBREAK;
        }
    }

    if((key != PAUSEBREAK) && (key != PRINTSCREEN)){
        if(code & FLAG_BREAK){
            make = 0;
        }
        else make = 1;
        keyRow = &keymap[(code & 0x7F) * MAP_COLS];
        column = 0;

        // 大小写
        int caps = shiftL || shiftR;
        if(capsLock && keyRow[0] >= 'a' && keyRow[0] <= 'z'){
            caps = !caps;
        }

        if(caps)   column = 1;
        if(codeE0) column = 2;

        key = keyRow[column];

        switch(key){
        case SHIFT_L: shiftL = make; break;
        case SHIFT_R: shiftR = make;break;

        case CTRL_L: ctrlL = make; break;
        case CTRL_R: ctrlR = make; break;

        case ALT_L: altL = make; break;
        case ALT_R: altR = make; break;

        case CAPS_LOCK:
            if(make){
                capsLock = !capsLock;
                setLeds();
            }
            break;
        case NUM_LOCK:
            if(make){
                numLock = !numLock;
                setLeds();
            }
            break;
        default: break;
        }
    }

    // pressed code
    if(make){
        int pad = 0;
```

```
            if((key >= PAD_SLASH) && (key <= PAD_9)){
                pad = 1;
                switch(key){
                case PAD_SLASH: key = '/';   break;
                case PAD_STAR:  key = '*';   break;
                case PAD_MINUS: key = '-';   break;
                case PAD_PLUS:  key = '+';   break;
                case PAD_ENTER: key = ENTER; break;
                default:
                    if(numLock){
                        if(key >= PAD_0 && key <= PAD_9){
                            key = key - PAD_0 + '0';
                        }
                        else if(key == PAD_DOT){
                            key = '.';
                        }
                    }
                    else{
                        switch(key){
                        case PAD_HOME:      key = HOME;     break;
                        case PAD_END:       key = END;      break;
                        case PAD_PAGEUP:    key = PAGEUP;   break;
                        case PAD_PAGEDOWN:  key = PAGEDOWN; break;
                        case PAD_UP:        key = UP;       break;
                        case PAD_DOWN:      key = DOWN;     break;
                        case PAD_LEFT:      key = LEFT;     break;
                        case PAD_RIGHT:     key = RIGHT;    break;
                        case PAD_DOT:       key = DELETE;   break;
                        case PAD_INS:       key = INSERT;   break;
                        default: break;
                        }
                    }
                    break;
                }
            }
            key |= shiftL   ? FLAG_SHIFT_L  : 0;
            key |= shiftR   ? FLAG_SHIFT_R  : 0;
            key |= ctrlL    ? FLAG_CTRL_L   : 0;
            key |= ctrlR    ? FLAG_CTRL_R   : 0;
            key |= altL     ? FLAG_ALT_L    : 0;
            key |= altR     ? FLAG_ALT_R    : 0;
            key |= pad      ? FLAG_PAD      : 0;

            in_process(tty, key);
        }
    }
}
```

- 处理完的键盘键入送往对应tty的in_process函数中进行最后处理，字符调用屏幕显示函数进行显示，功能按键调用其他模块或系统的对应功能

3.4 进程管理部分

- 进程调度：根据进程的优先级进行调度

```
PUBLIC void schedule()
```

```
{
    struct proc*    p;
    int     greatest_ticks = 0;

    while (!greatest_ticks) {
        for (p = &FIRST_PROC; p <= &LAST_PROC; p++) {
            if (p->p_flags == 0) {
                if (p->ticks > greatest_ticks) {
                    greatest_ticks = p->ticks;
                    p_proc_ready = p;
                }
            }
        }

        if (!greatest_ticks)
            for (p = &FIRST_PROC; p <= &LAST_PROC; p++)
                if (p->p_flags == 0)
                    p->ticks = p->priority;
    }
}
```

- 进程结构

```
struct proc {
    struct stackframe regs;    /* process registers saved in stack frame */

    u16 ldt_sel;               /* gdt selector giving ldt base and limit */
    struct descriptor ldts[LDT_SIZE]; /* local descs for code and data */

        int ticks;                 /* remained ticks */
        int priority;

    u32 pid;                   /* process id passed in from MM */
    char name[16];         /* name of the process */

    int  p_flags;              /**
                    * process flags.
                    * A proc is runnable iff p_flags==0
                    */
    int run_state;          /*a proc is running if run_state==1*/
    MESSAGE * p_msg;
    int p_recvfrom;
    int p_sendto;

    int has_int_msg;           /**
                    * nonzero if an INTERRUPT occurred when
                    * the task is not ready to deal with it.
                    */

    struct proc * q_sending;   /**
                    * queue of procs sending messages to
                    * this proc
                    */
    struct proc * next_sending;/**
                    * next proc in the sending
                    * queue (q_sending)
                    */
    /* int nr_tty; */

    struct file_desc * filp[NR_FILES];
};
```

3.5 内存管理（子进程生成）部分

核心代码：
//分配进程表->分配内存->生成子进程，完成共享
PUBLIC int mm_fork()
{
    struct proc* p = proc_table;
    int i;
    for (i = 0; i < NR_TASKS + NR_PROCS; i++,p++)
        if (p->p_flags == FREE_SLOT)
            break;

```
int child_pid = i;
assert(p == &proc_table[child_pid]);
assert(child_pid >= NR_TASKS + NR_NATIVE_PROCS);
if (i == NR_TASKS + NR_PROCS)
    return -1;
assert(i < NR_TASKS + NR_PROCS);

int pid = mm_msg.source;
printl("{MM} pid: %d\n", pid);
u16 child_ldt_sel = p->ldt_sel;
*p = proc_table[pid];
p->ldt_sel = child_ldt_sel;
p->p_parent = pid;
p->pid = child_pid;
sprintf(p->name, "%s_%d", proc_table[pid].name, child_pid);

struct descriptor * ppd;

ppd = &proc_table[pid].ldts[INDEX_LDT_C];
printl("{MM} name: %s\n", proc_table[pid].name);
int caller_T_base  = reassembly(ppd->base_high, 24,
                ppd->base_mid,  16,
                ppd->base_low);
int caller_T_limit = reassembly(0, 0,
                (ppd->limit_high_attr2 & 0xF), 16,
                ppd->limit_low);
int caller_T_size  = ((caller_T_limit + 1) *
              ((ppd->limit_high_attr2 & (DA_LIMIT_4K >> 8)) ?
               4096 : 1));
printl("{MM} %x,%x,%x,%x,%x,%x)\n", ppd->limit_low, ppd->base_low,  ppd-
>base_high, ppd->base_mid, ppd->attr1, ppd->limit_high_attr2);
printl("{MM} %x\n", caller_T_limit+1);
printl("{MM} %x\n", (ppd->limit_high_attr2 & (DA_LIMIT_4K >> 8)));
printl("{MM} %x\n", ((ppd->limit_high_attr2 & (DA_LIMIT_4K >> 8))? 4096:1));
printl("{MM} %x\n", (caller_T_limit+1)*((ppd->limit_high_attr2 & (DA_LIMIT_4K >>
8))? 4096:1));
```

```
printl("{MM} %x\n", ((caller_T_limit+1)>>10)*((ppd->limit_high_attr2 &
(DA_LIMIT_4K >> 8))? 4096:1));

ppd = &proc_table[pid].ldts[INDEX_LDT_RW];
printl("{MM} %x,%x,%x,%x,%x,%x)\n", ppd->limit_low, ppd->base_low,  ppd-
>base_high, ppd->base_mid, ppd->attr1, ppd->limit_high_attr2);
int caller_D_S_base  = reassembly(ppd->base_high, 24,
                  ppd->base_mid,  16,
                  ppd->base_low);
int caller_D_S_limit = reassembly((ppd->limit_high_attr2 & 0xF), 16,
                  0, 0,
                  ppd->limit_low);
int caller_D_S_size  = ((caller_T_limit + 1) *
            ((ppd->limit_high_attr2 & (DA_LIMIT_4K >> 8)) ?
             4096 : 1));

printl("{MM} base: %d, limit: %d, size: %d)\n", caller_T_base, caller_T_limit,
caller_T_size);
assert((caller_T_base  == caller_D_S_base ) &&
       (caller_T_limit == caller_D_S_limit) &&
       (caller_T_size  == caller_D_S_size ));

caller_T_size = caller_D_S_size = 0x100000;
int child_base = alloc_mem(child_pid, caller_T_size);
printl("{MM} childpid:%x  0x%x <- 0x%x (0x%x bytes)\n",
        child_pid, child_base, caller_T_base, caller_T_size);
phys_copy((void*)child_base, (void*)caller_T_base, caller_T_size);

init_descriptor(&p->ldts[INDEX_LDT_C],
      child_base,
      (PROC_IMAGE_SIZE_DEFAULT - 1) >> LIMIT_4K_SHIFT,
      DA_LIMIT_4K | DA_32 | DA_C | PRIVILEGE_USER << 5);
init_descriptor(&p->ldts[INDEX_LDT_RW],
      child_base,
      (PROC_IMAGE_SIZE_DEFAULT - 1) >> LIMIT_4K_SHIFT,
      DA_LIMIT_4K | DA_32 | DA_DRW | PRIVILEGE_USER << 5);

mm_msg.PID = child_pid;

MESSAGE m;
m.type = SYSCALL_RET;
m.RETVAL = 0;
m.PID = 0;
send_recv(SEND, child_pid, &m);

return 0;
```

}

3.6 游戏2048

- 该游戏通过左右上下移动函数作为主体驱动游戏

```
static void move_left();  /* 左移 */
static void move_right(); /* 右移 */
static void move_up();    /* 上移 */
static void move_down();  /* 下移 */
```

3.7 游戏井字棋

核心代码（AI部分）：

```
void computer()
{
    int i;
    if (arrfull())
    {
        for (i = 1; i <= 9; i++)
        {
            if (i == arr[i] - 48)
            {
                c3 = 0; n2 = 0; c2 = 0; n1 = 0; c1 = 0;
                arr[i] = 'X';
                number = linenum(arr[1], arr[2], arr[3]); cn(number);
                number = linenum(arr[4], arr[5], arr[6]); cn(number);
                number = linenum(arr[7], arr[8], arr[9]); cn(number);
                number = linenum(arr[1], arr[4], arr[7]); cn(number);
                number = linenum(arr[2], arr[5], arr[8]); cn(number);
                number = linenum(arr[3], arr[6], arr[9]); cn(number);
                number = linenum(arr[1], arr[5], arr[9]); cn(number);
                number = linenum(arr[3], arr[5], arr[7]); cn(number);
                brr[i] = (128 * c3 - 63 * n2 + 31 * c2 - 15 * n1 + 7 * c1);
                arr[i] = i + 48;
            }
            else
                brr[i] = -999;
        }
        arr[maxbrr(brr)] = 'X';
        c3 = 0; n2 = 0; c2 = 0; n1 = 0; c1 = 0;
        number = linenum(arr[1], arr[2], arr[3]); cn(number);
        number = linenum(arr[4], arr[5], arr[6]); cn(number);
        number = linenum(arr[7], arr[8], arr[9]); cn(number);
        number = linenum(arr[1], arr[4], arr[7]); cn(number);
        number = linenum(arr[2], arr[5], arr[8]); cn(number);
        number = linenum(arr[3], arr[6], arr[9]); cn(number);
        number = linenum(arr[1], arr[5], arr[9]); cn(number);
        number = linenum(arr[3], arr[5], arr[7]); cn(number);
        if (c3 != 0)
        {
            display(arr);
            printf("\n");
            printf("PC win!!!\n");
```

```
        suc = 0;
    }
}
else
    suc = 0;
```

}

3.8 游戏算排列

核心代码：

```
void recall(int l)
{
    int i, j;
    if (l == n + 1)
    {
        x = x + 1;
        printf("stacking methods are:\n", x);
        for (i = 1; i <= n; i++)
            printf("%d", h[i]);
        printf("\n");
    }
    for (i = 1; i <= n; i++)
    {
        h[l] = i;
        if (jc(l, i) != 1)
            huisu(l + 1);
    }
}
```

3.9 游戏计算器

核心代码：

```
switch(bufr[1])
{
    case '+':result=chartonumber(bufr[0])+chartonumber(bufr[2]);break;
    case '-':result=chartonumber(bufr[0])-chartonumber(bufr[2]);break;
    case '':result=chartonumber(bufr[0])chartonumber(bufr[2]);break;
    case '/':result=chartonumber(bufr[0])/chartonumber(bufr[2]);break;
}
```

3.10 游戏猜数字

核心代码：

```
while (b<a)
{
    printf("Too low.Try again.\n");
    read(0, bufr, 128);
    b=my_atoi(bufr);
}
while (b>a)
{
    printf("Too high.Try again.\n");
    read(0, bufr, 128);
    b=my_atoi(bufr);
}
```

3.11 游戏双人五子棋

核心代码：
int GobangJudge(int x, int y)
{
    int i, j;
    int t = 2 - whoseTurn % 2;

```
for (i = x - 4, j = y; i <= x; i++)
{
    if (i >= 1 && i <= N - 4 && t == chessboard[i][j] && t == chessboard[i + 1]
[j] && t == chessboard[i + 2][j] && t == chessboard[i + 3][j] && t ==
chessboard[i + 4][j])
        return 1;
}
for (i = x, j = y - 4; j <= y; j++)
{
    if (j >= 1 && j <= N - 4 && t == chessboard[i][j] && t == chessboard[i][j +
1] && t == chessboard[i][j + 1] && t == chessboard[i][j + 3] && t ==
chessboard[i][j + 4])
        return 1;
}
for (i = x - 4, j = y - 4; i <= x, j <= y; i++, j++)
{
    if (i >= 1 && i <= N - 4 && j >= 1 && j <= N - 4 && t == chessboard[i][j] &&
t == chessboard[i + 1][j + 1] && t == chessboard[i + 2][j + 2] && t ==
chessboard[i + 3][j + 3] && t == chessboard[i + 4][j + 4])
        return 1;
}
for (i = x + 4, j = y - 4; i >= 1, j <= y; i--, j++)
{
    if (i >= 1 && i <= N - 4 && j >= 1 && j <= N - 4 && t == chessboard[i][j] &&
t == chessboard[i - 1][j + 1] && t == chessboard[i - 2][j + 2] && t ==
chessboard[i - 3][j + 3] && t == chessboard[i - 4][j + 4])
        return 1;
}

return 0;
```

}

3.12游戏推箱子

核心代码：

对于地图的定义：

void draw_map(int map[9][11])
{
    int i;
    int j;
    for (i = 0; i < 9; i++)
    {
        for (int j = 0; j < 11; j++)
        {
            switch (map[i][j])
            {
            case 0:

```
                printf(" "); //道路
                break;
            case 1:
                printf("#"); //墙壁
                break;
            case 2:
                printf(" "); //游戏边框的空白部分
                break;
            case 3:
                printf("D"); //目的地
                break;
            case 4:
                printf("b"); //箱子
                break;
            case 5:
                printf("!"); //箱子进入目的地
                break;
            case 6:
                printf("p"); //人
                break;
            case 7:
                printf("^"); //人进入目的地
                break;
            }
        }
        printf("\n");
    }
}
```

对于各种情况的判断以及实现：

```
case 'w':
            //如果人前面是空地。
            if (map[pi - 1][pj] == 0)
            {
                map[pi - 1][pj] = 6 + 0;
                if (map[pi][pj] == 9)
                    map[pi][pj] = 3;
                else
                    map[pi][pj] = 0;
            }
            //如果人前面是目的地。
            else if ((map[pi - 1][pj] == 3) || (map[pi - 1][pj] == 9))
            {
                map[pi - 1][pj] = 6 + 3;
                if (map[pi][pj] == 9)
                    map[pi][pj] = 3;
                else
                    map[pi][pj] = 0;
            }
            //如果人前面是箱子。
            else if (map[pi - 1][pj] == 4)
            {
```

```
        if (map[pi - 2][pj] == 0)
        {
            map[pi - 2][pj] = 4;
            if (map[pi - 1][pj] == 7)
                map[pi - 1][pj] = 9;
            else
                map[pi - 1][pj] = 6;
            if (map[pi][pj] == 9)
                map[pi][pj] = 3;
            else
                map[pi][pj] = 0;
        }
break;
```