

MiniOS —— 操作系统课程设计文档



项目成员：

- 1851197 周楷彬 42028701
- 1851352 伏瑞 42028702
- 1851199 庄泽杨 42028703
- 1851494 王宇其 42028701
- 1851198 吕嘉尧 42028701

指导老师：王冬青老师

目录

MiniOS —— 操作系统课程设计文档

目录

1.项目概述

- 1.1 项目简介
- 1.2 开发环境
- 1.3 项目分工

2.操作指南

3.功能展示

- 3.1 图形界面
- 3.2 系统级应用——多用户登录
- 3.3 系统级应用——进程管理
- 3.4 系统级应用——多功能控制台
- 3.5 系统级应用——文件管理
- 3.6 用户级应用——画图
- 3.7 用户级应用——计算器
- 3.8 用户级应用——日历
- 3.9 用户级应用——2048
- 3.10 用户级应用——五子棋

- 3.11 用户级应用——猜数字
- 3.12 用户级应用——拼图
- 3.13 用户级应用——推箱子
- 3.14 用户级应用——MoveGame
- 3.15 用户级应用——贪吃蛇
- 4.功能实现
 - 4.1 图形界面
 - 4.2 系统级应用——多用户登录
 - 4.3 系统级应用——进程管理
 - 4.4 系统级应用——多功能控制台
 - 4.5 系统级应用——文件管理
 - 4.6 用户级应用——画图
 - 4.7 用户级应用——计算器
 - 4.8 用户级应用——日历
 - 4.9 用户级应用——2048
 - 4.10 用户级应用——五子棋
 - 4.11 用户级应用——猜数字
 - 4.12 用户级应用——拼图
 - 4.13 用户级应用——推箱子
 - 4.14 用户级应用——MoveGame
 - 4.15 用户级应用——贪吃蛇

1.项目概述

1.1 项目简介

1. 我们的项目基于Orange's的样例代码，主要参考资料为《Orange'S 一个操作系统的实现》，在上面进行了添加，删除和修改，同时参考了《30天自制操作系统》
2. 项目的命令行系统参考自linux，我们对其进行了模仿
3. 在样例代码的基础上，我们的项目实现了 320x200x8 位彩色的图形模式，增加了**图形管理模块**，能够读取并显示BMP格式图片，并提供了一组绘图接口，增加了**4个系统级应用**，包括多用户登录、进程管理、多功能控制台、文件管理。同时，还实现了**10个用户级应用**，包括画图、计算器、日历和7个小游戏（2048，五子棋，猜数字，拼图，推箱子，MoveGame，贪吃蛇）

1.2 开发环境

- 平台：ubuntu 14.04.5 (32bit)
- 语言：汇编语言，C语言
- 工具：gcc, nasm,make,bochs 2.6.11

1.3 项目分工

- 周楷彬：图形管理模块，多用户登录，画图，代码整合
- 伏瑞：进程管理，推箱子，猜数字
- 庄泽杨：文件管理，推箱子，五子棋
- 王宇其：多功能控制台，拼图，MoveGame
- 吕嘉尧：计算器，日历，2048，贪吃蛇

2.操作指南

命令	说明
help	显示所有命令
clear	清屏
colormode	切换颜色模式(彩色/灰阶)
bmp	显示BMP图片
logout	登出
sudo	切换到管理员用户
users	显示所有用户
useradd [username] [password]	添加用户 <code>username</code> ,密码为 <code>password</code>
userdel [username]	删除用户 <code>username</code>
passwd [username]	修改用户 <code>username</code> 的密码
proc	显示所有进程
kill [pid]	结束pid值为 <code>pid</code> 的进程
pause [pid]	暂停pid值为 <code>pid</code> 的进程
resume [pid]	恢复pid值为 <code>pid</code> 的进程
↑↓键	切换执行过的历史命令
<code>TAB</code> 键	命令智能补全
ls	显示所有文件
mkfile [filename]	创建文件 <code>filename</code>
cat [filename]	读取并显示文件 <code>filename</code>
rm [filename]	删除文件 <code>filename</code>
wt [filename]	编辑文件 <code>filename</code> ,覆盖其原来的内容
wt+ [filename]	编辑文件 <code>filename</code> ,追加内容
draw	运行画图程序
calculator	运行计算器程序
calendar	运行日历程序
2048	运行2048游戏
chess	运行五子棋游戏
guessnum	运行猜数字游戏
pintu	运行拼图游戏
pushbox	运行推箱子游戏

命令	说明
movegame	运行MoveGame游戏
tanchishe	运行贪吃蛇游戏

3.功能展示

3.1 图形界面

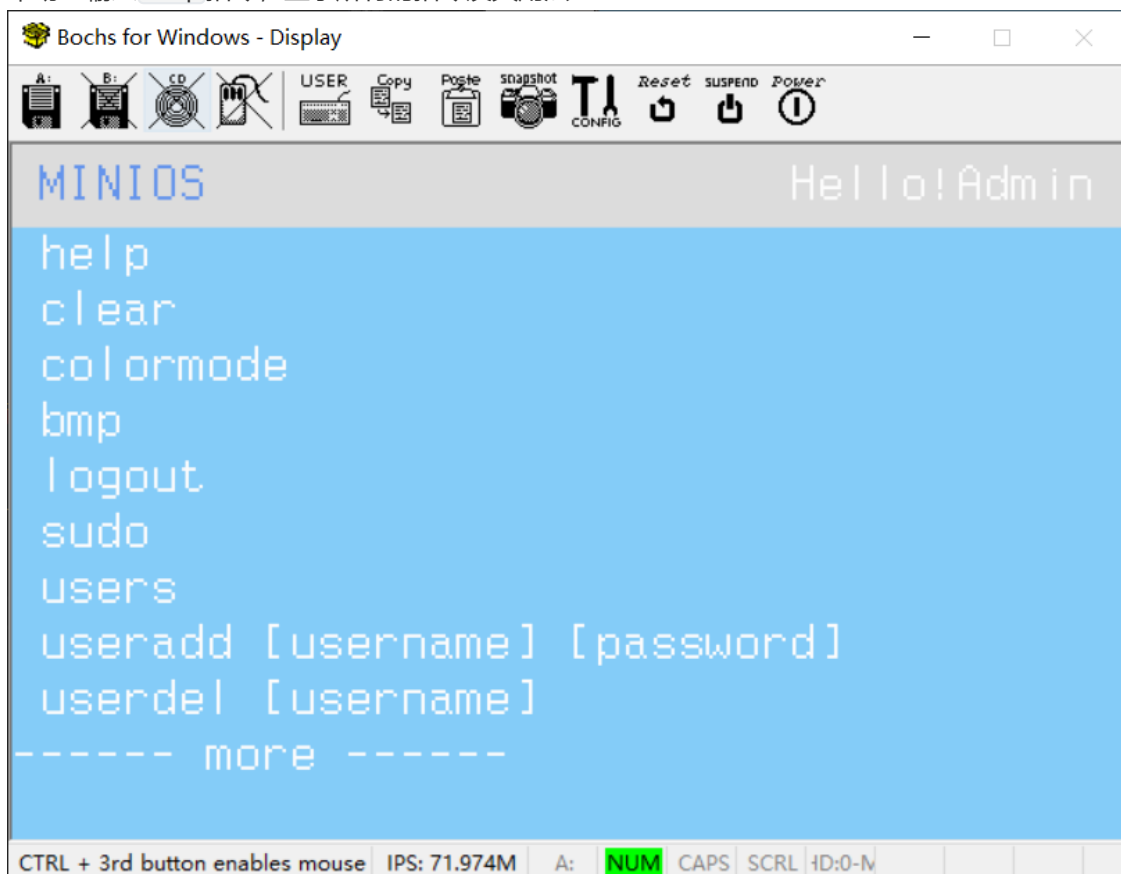
- 运行操作系统，完成加载工作后，进入图形模式，显示开机画面后，进入登录界面，初始状态只有管理员账户 Admin，密码为 admin

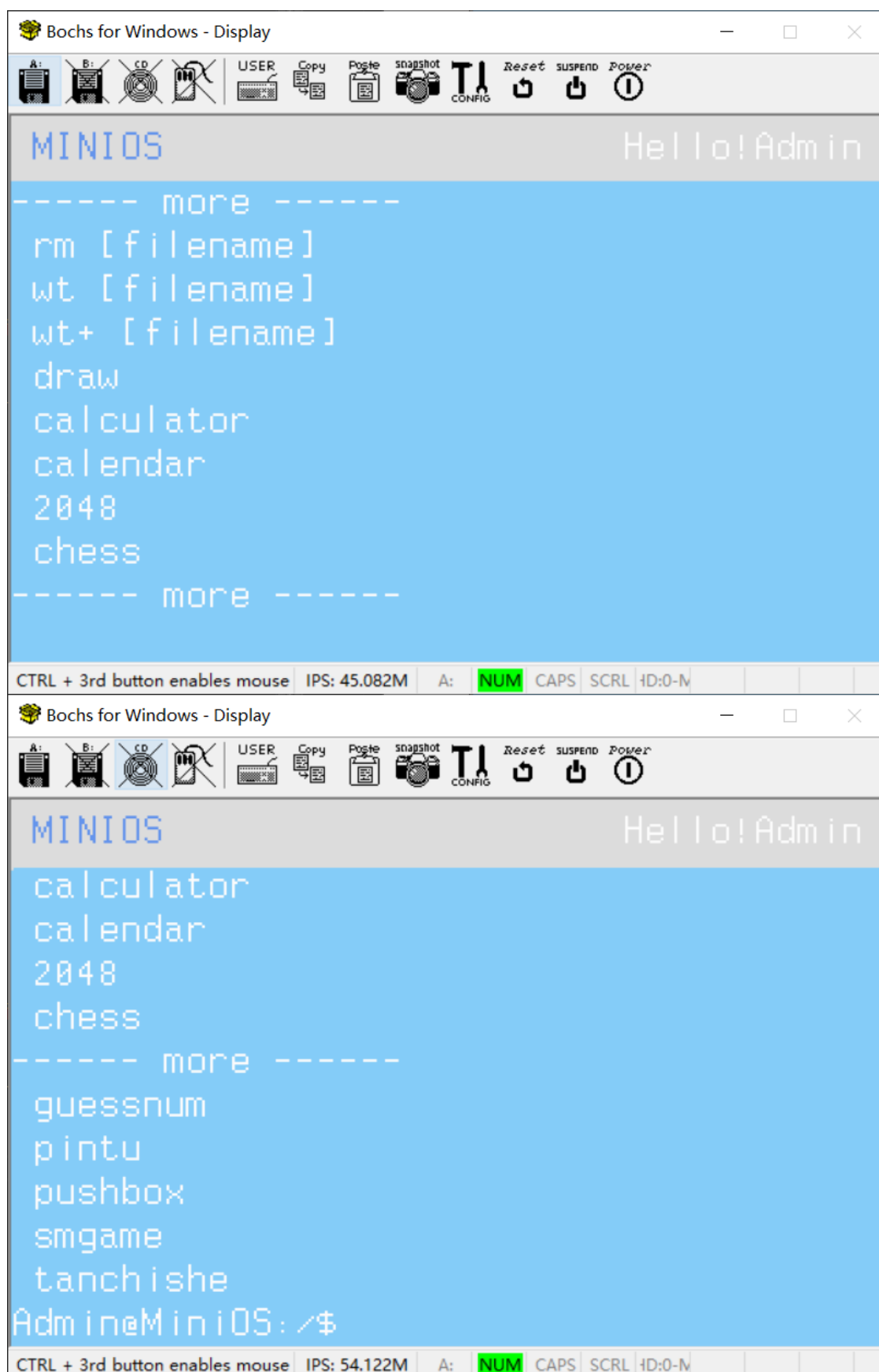


- 输入正确密码后成功登录，进入主界面，在控制台输入指令后，操作系统就会执行相应的程序



- 帮助：输入 help 指令，显示所有的指令及其用法





- 清屏：输入 `clear` 指令，清空控制台输出



- 切换颜色模式：输入 `colormode` 指令，系统会切换图形模式下使用的色盘，从而切换颜色模式(彩色/灰阶)



- 显示图片：输入 bmp 指令，系统会读取并显示一张使用8位索引颜色模式的BMP格式图片



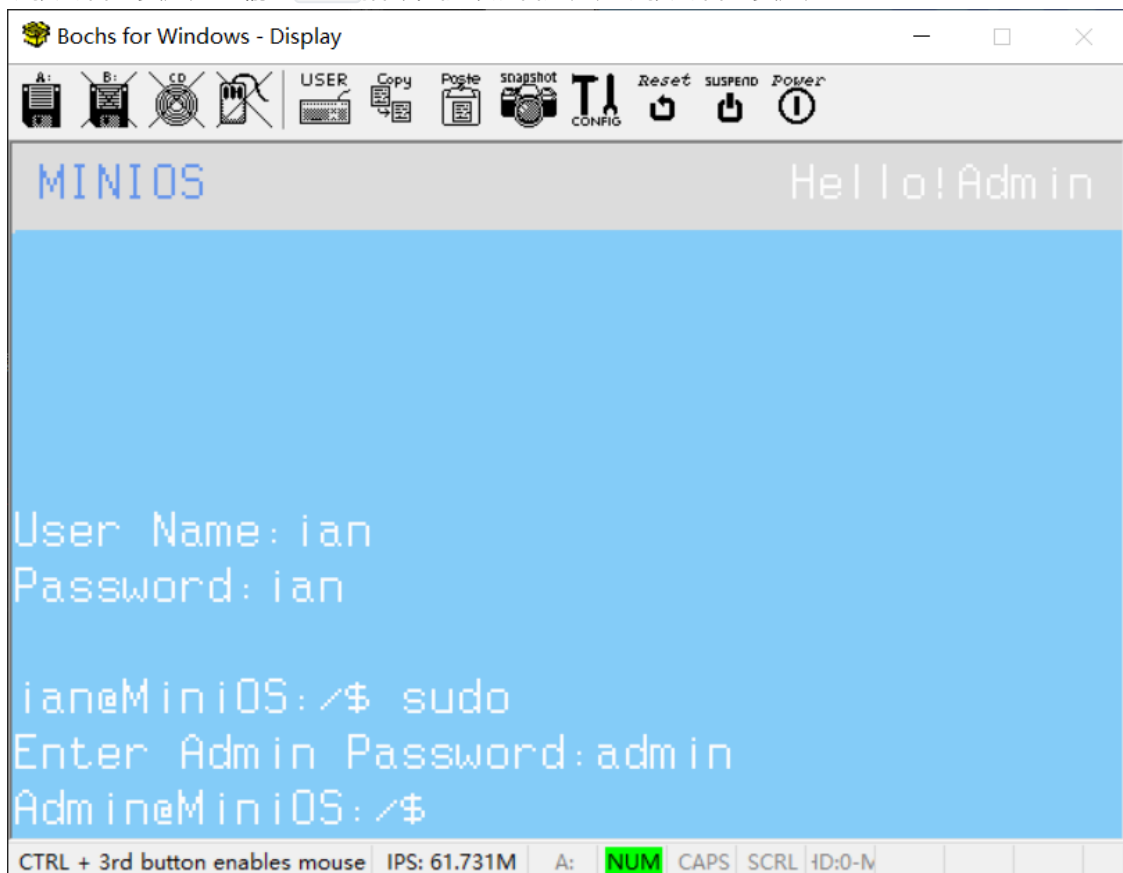
3.2 系统级应用——多用户登录

我们的系统增加了多用户登录及权限功能，不同的用户有不同的权限和自己的文件分区，除了管理员用户，普通用户只能访问和修改自己的文件，只能修改自己的密码

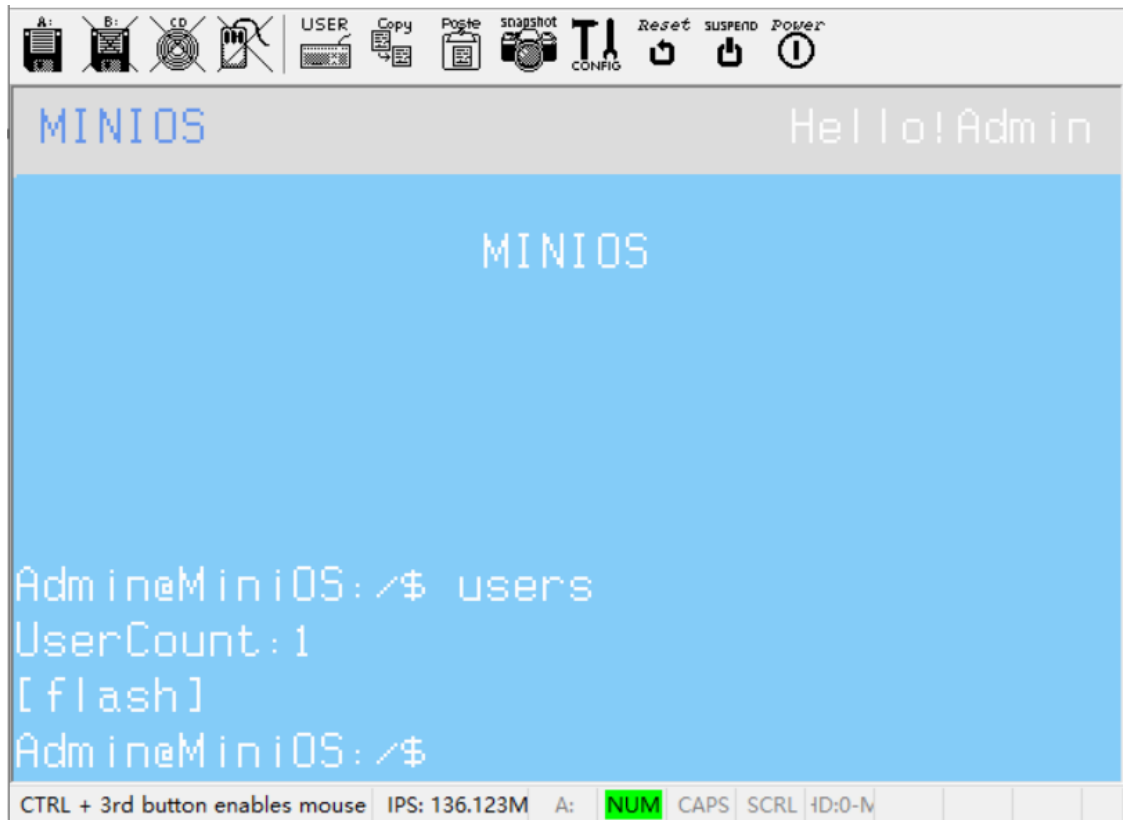
- 登出：输入 `Logout` 指令，退出登录，系统要求重新登录



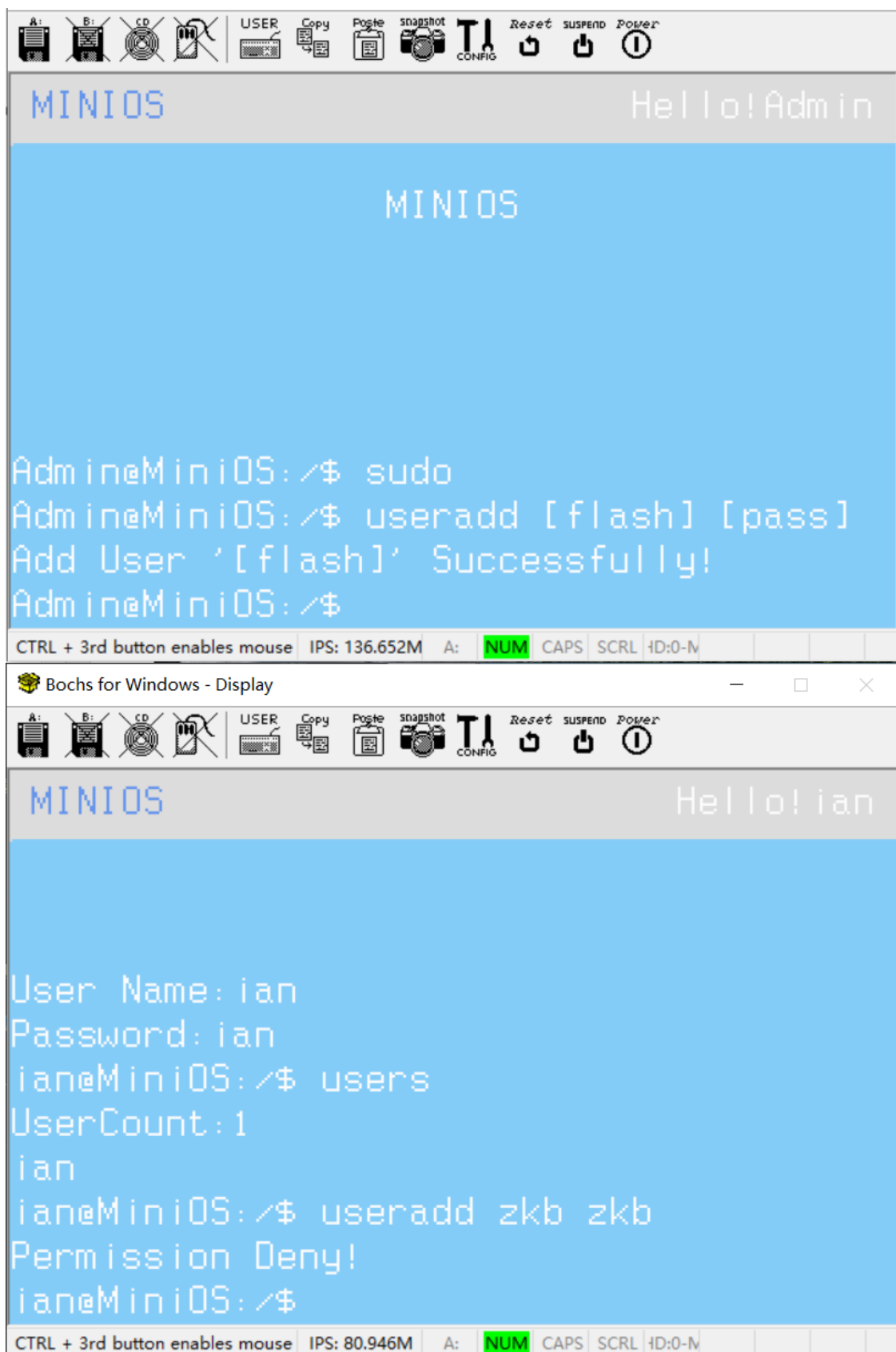
- 切换到管理员用户：输入 `sudo` 指令，能够从普通用户切换到管理员用户



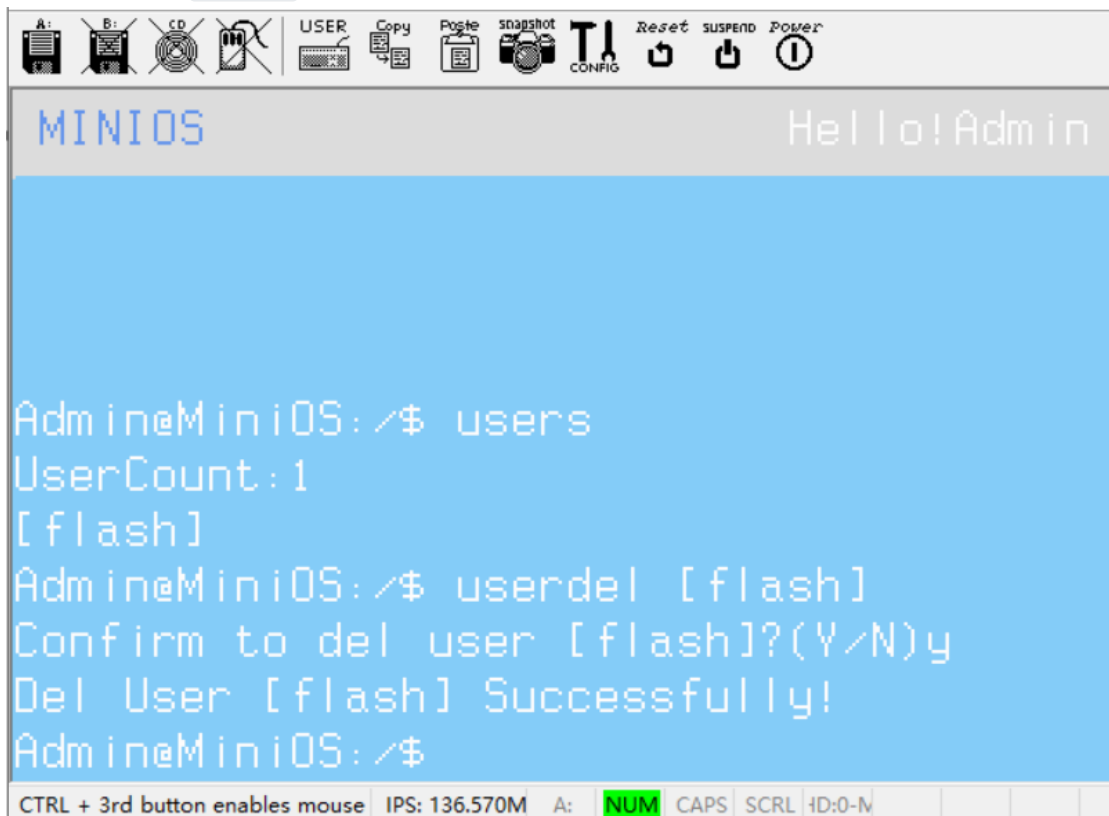
- 显示所有用户：输入 `users` 指令，显示系统中的用户数量和所有用户名



- 添加用户：输入 `useradd` 指令，添加相应的用户，只有管理员用户能够添加用户，普通用户要求添加用户会输出没有权限的错误



- 删除用户：输入 `userdel` 指令，删除相应的用户，只有管理员用户能够删除用户

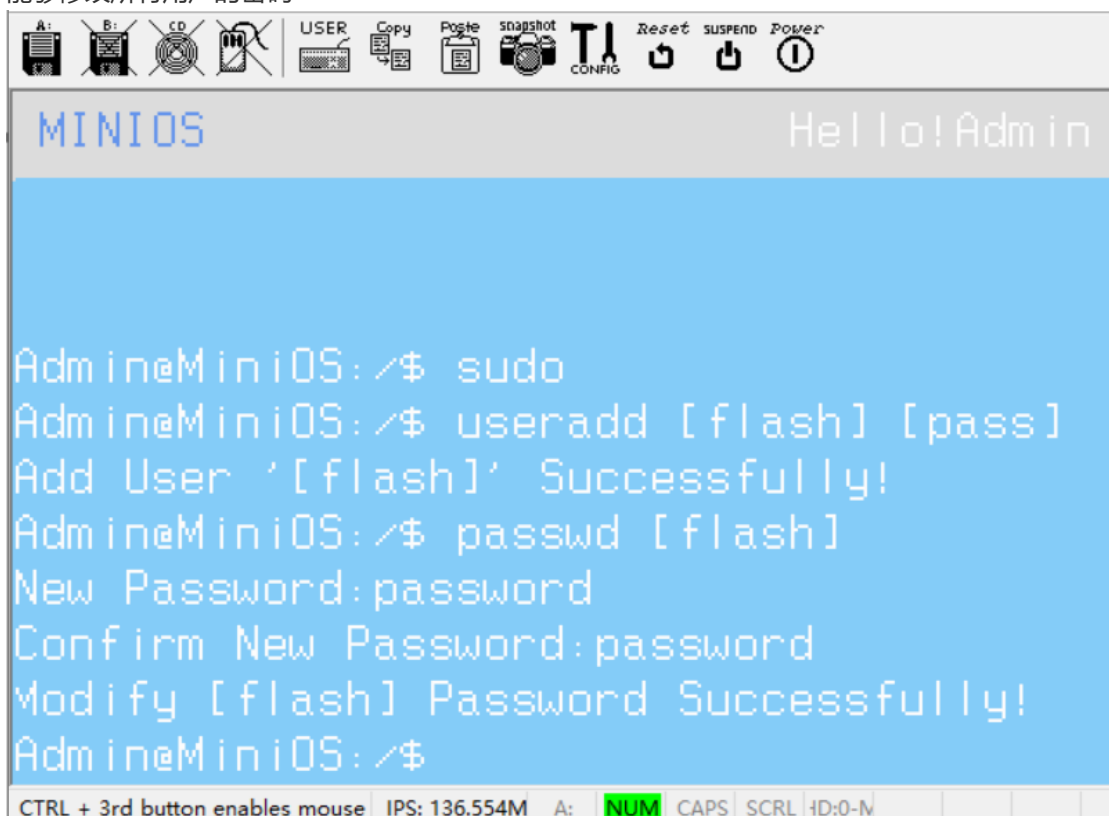


The screenshot shows the MINIOS terminal interface. At the top, there is a toolbar with icons for A:, B:, CD, USER, Copy, Poste, snapshot, CONFIG, Reset, SUSPEND, and Power. Below the toolbar, the text "MINIOS" is on the left and "Hello! Admin" is on the right. The main area is a blue terminal window. The prompt is "Admin@MiniOS:/\$". The user enters "users", and the output is "UserCount:1". The user enters "[flash]". The prompt is "Admin@MiniOS:/\$". The user enters "userdel [flash]". The output is "Confirm to del user [flash]?(Y/N)y". The output is "Del User [flash] Successfully!". The prompt is "Admin@MiniOS:/\$". At the bottom, there is a status bar with the text "CTRL + 3rd button enables mouse", "IPS: 136.570M", "A: NUM", "CAPS", "SCRL", and "ID:0-M".

```

Admin@MiniOS:/$ users
UserCount:1
[flash]
Admin@MiniOS:/$ userdel [flash]
Confirm to del user [flash]?(Y/N)y
Del User [flash] Successfully!
Admin@MiniOS:/$
  
```

- 修改密码：输入 `passwd` 指令，修改相应用户的密码，普通用户只能修改自己的密码，管理员用户能够修改所有用户的密码



The screenshot shows the MINIOS terminal interface. At the top, there is a toolbar with icons for A:, B:, CD, USER, Copy, Poste, snapshot, CONFIG, Reset, SUSPEND, and Power. Below the toolbar, the text "MINIOS" is on the left and "Hello! Admin" is on the right. The main area is a blue terminal window. The prompt is "Admin@MiniOS:/\$". The user enters "sudo". The prompt is "Admin@MiniOS:/\$". The user enters "useradd [flash] [pass]". The output is "Add User '[flash]' Successfully!". The prompt is "Admin@MiniOS:/\$". The user enters "passwd [flash]". The output is "New Password:password". The output is "Confirm New Password:password". The output is "Modify [flash] Password Successfully!". The prompt is "Admin@MiniOS:/\$". At the bottom, there is a status bar with the text "CTRL + 3rd button enables mouse", "IPS: 136.554M", "A: NUM", "CAPS", "SCRL", and "ID:0-M".

```

Admin@MiniOS:/$ sudo
Admin@MiniOS:/$ useradd [flash] [pass]
Add User '[flash]' Successfully!
Admin@MiniOS:/$ passwd [flash]
New Password:password
Confirm New Password:password
Modify [flash] Password Successfully!
Admin@MiniOS:/$
  
```

3.3 系统级应用——进程管理

- proc

MINIOS Hello! Admin

1	SYS	running	-
2	HD	running	-
3	FS	running	-
4	MM	running	-
5	GRAPHICS	running	-
6	INIT	running	-
7	TestA	running	
8	TestB	running	
9	TestC	running	

Admin@MinIOS:/\$ proc

CTRL + 3rd button enables mouse IPS: 135.437M A: NUM CAPS SCRL HD:0-M

- kill

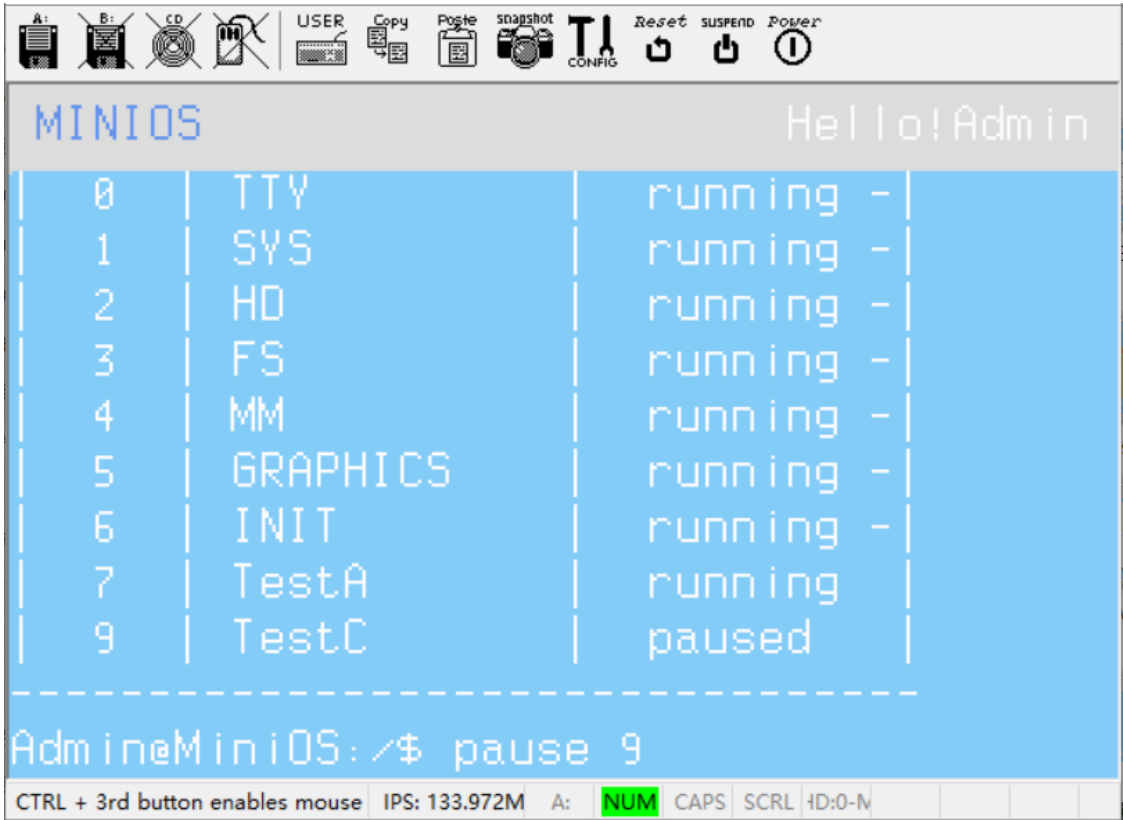
MINIOS Hello! Admin

0	TTY	running	-
1	SYS	running	-
2	HD	running	-
3	FS	running	-
4	MM	running	-
5	GRAPHICS	running	-
6	INIT	running	-
7	TestA	running	
9	TestC	running	

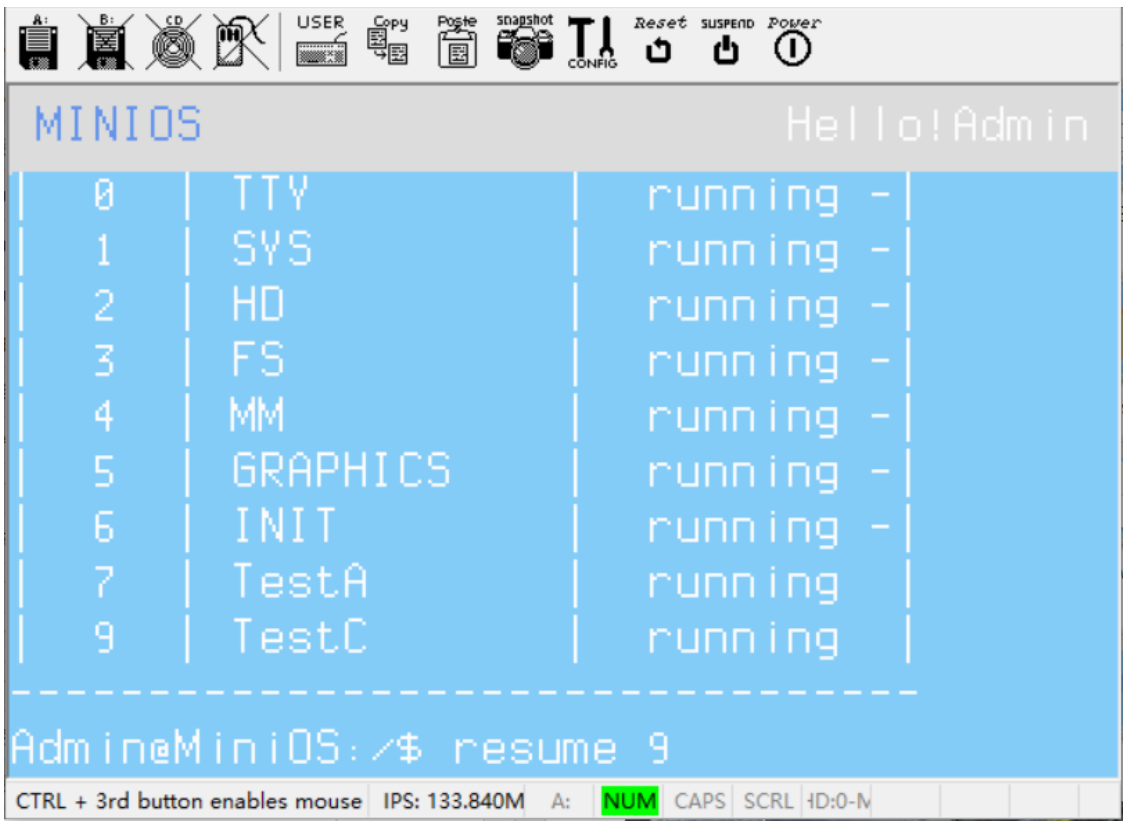
Admin@MinIOS:/\$ kill 8

CTRL + 3rd button enables mouse IPS: 132.478M A: NUM CAPS SCRL HD:0-M

- pause



- resume



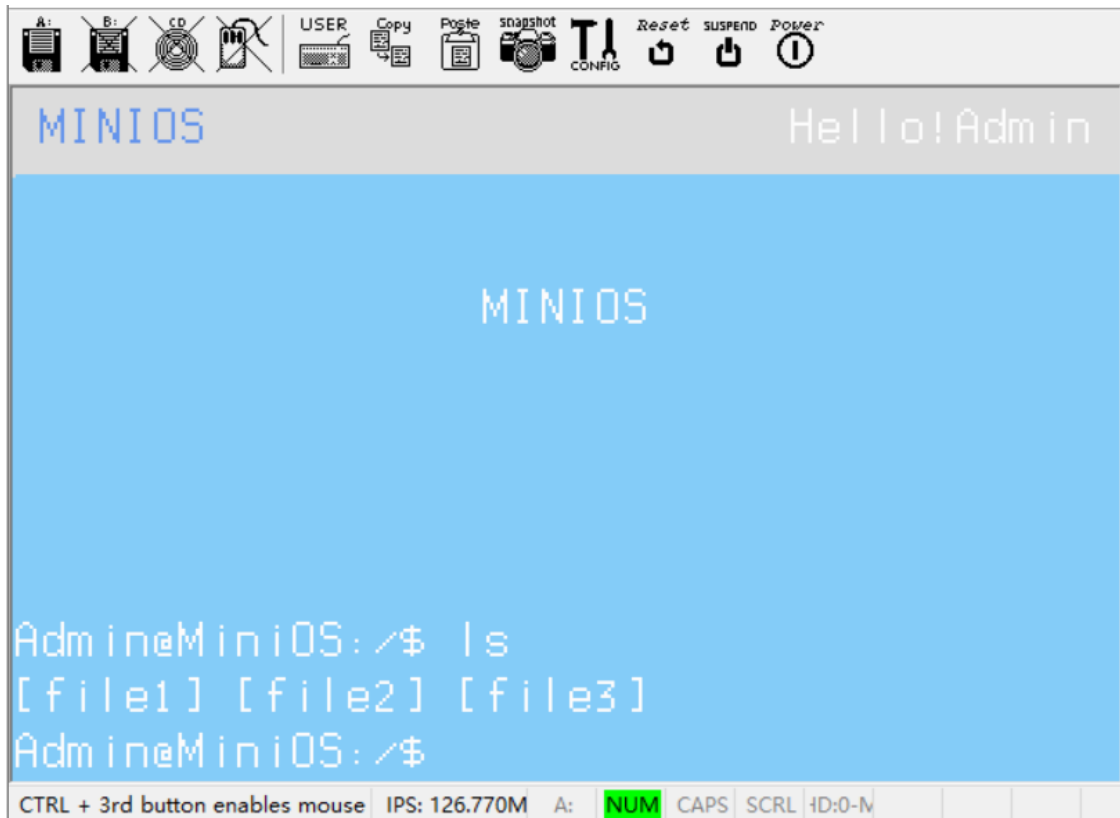
3.4 系统级应用——多功能控制台

- ↑键: '↑'和'↓'可以查看上（下）一条运行的指令
- TAB 键: 智能补全指令

3.5 系统级应用——文件管理

我们将Orange'S原本的扁平化文件系统改为了多用户分区的文件系统，不同用户拥有不同分区，且只能访问自己的分区，在各自分区下的文件系统进行各自的文件系统操作，具体操作如下：

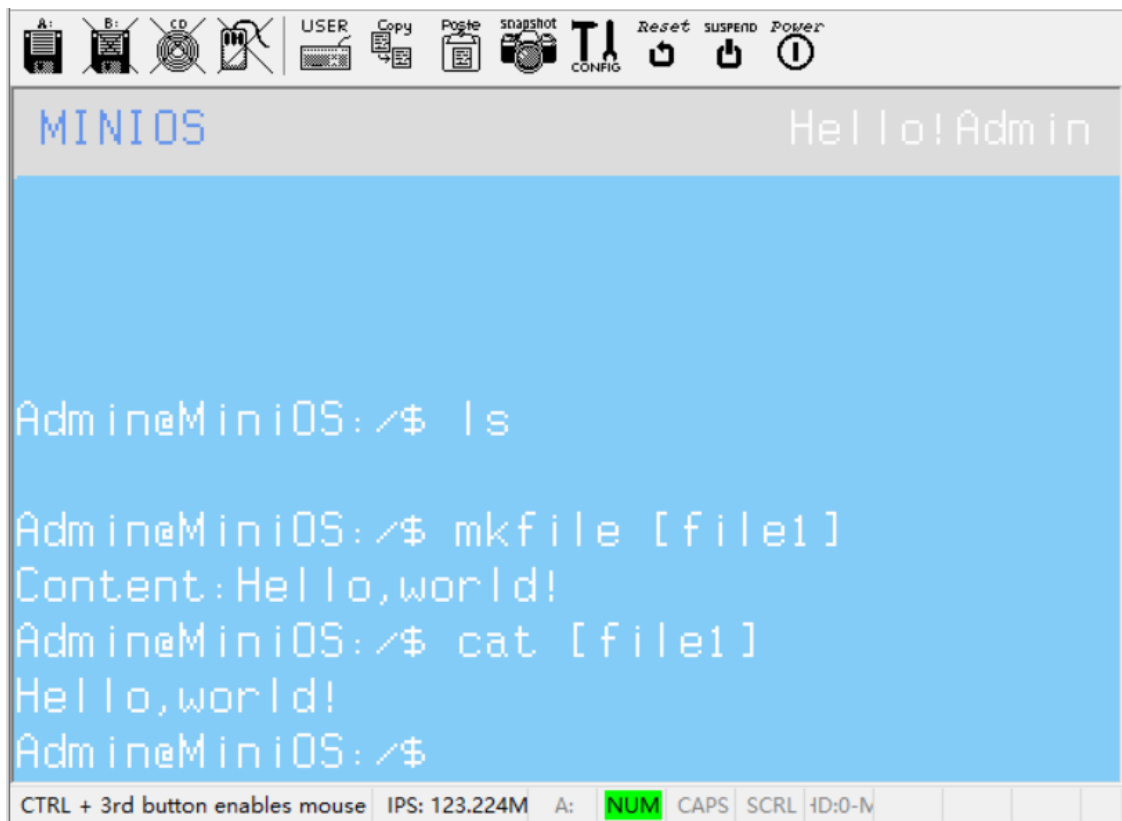
- 显示所有文件：使用 `ls` 指令查看当前目录下的所有文件



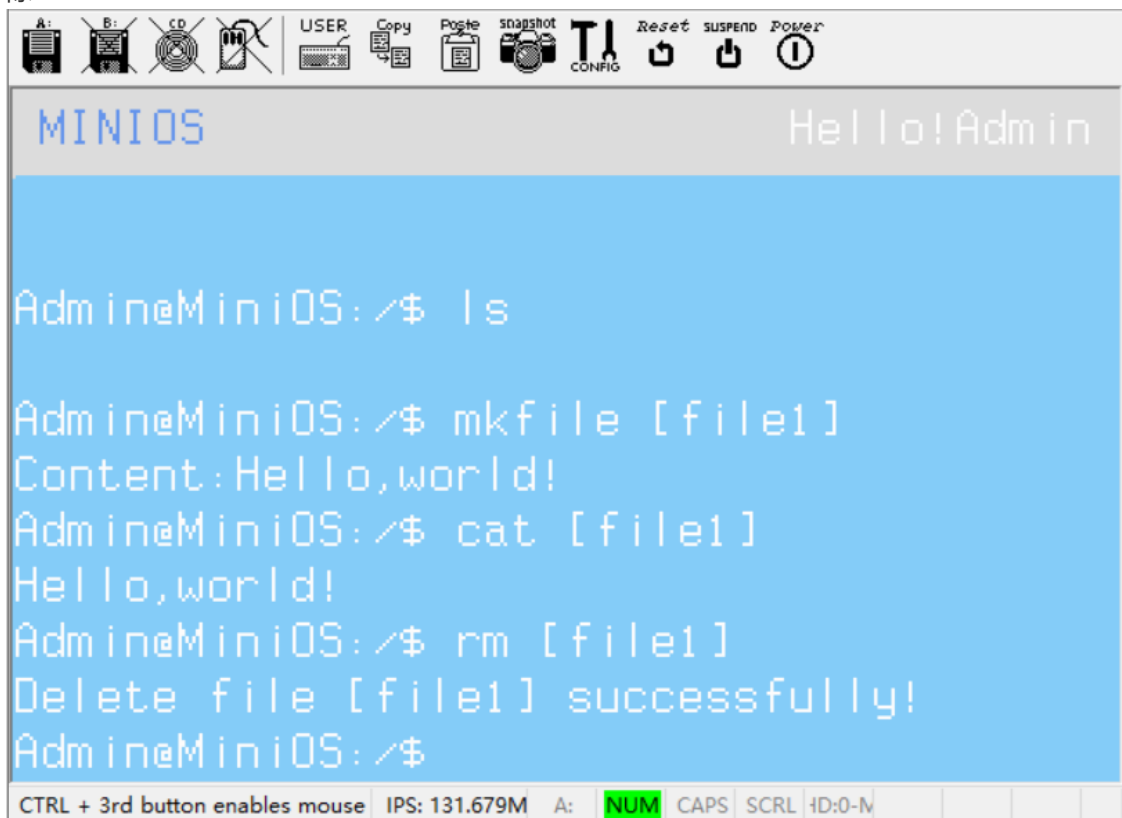
- 创建文件：输入 `mkfile [filename]` 指令创建文件名为 `filename` 的文件，且需输入该文件的初始内容



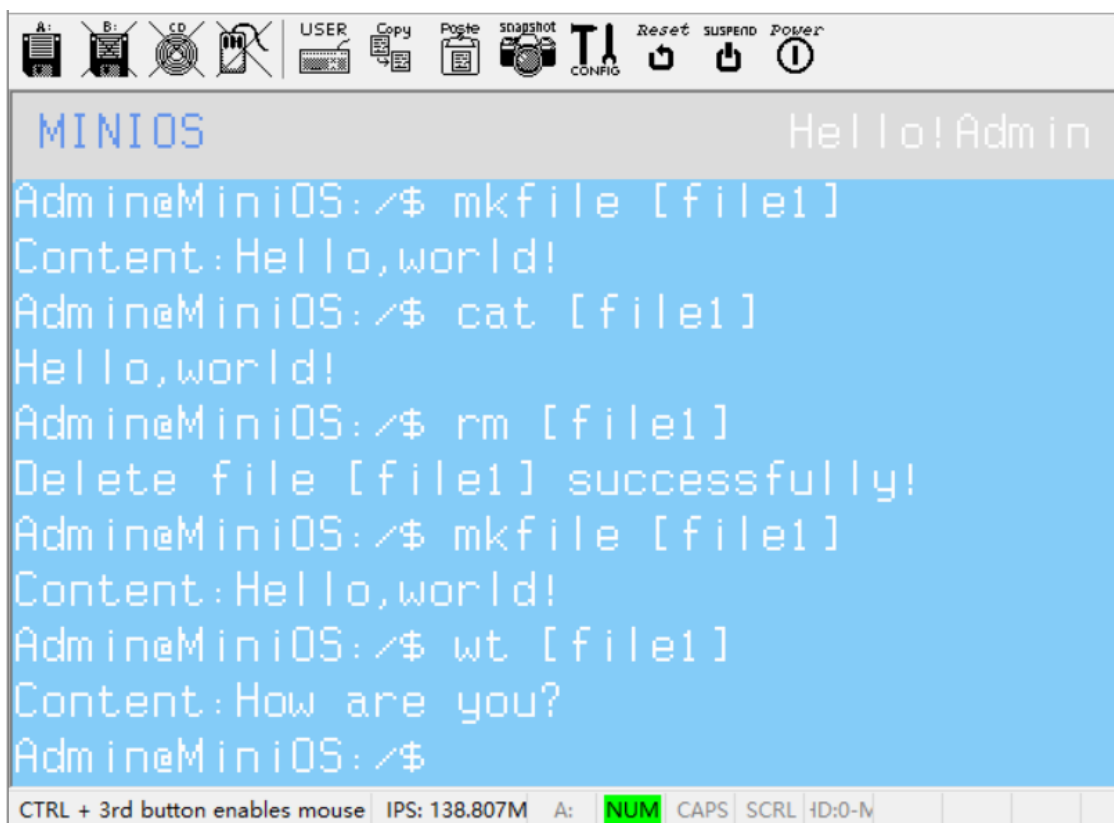
- 读取并显示文件：输入 `cat [filename]` 指令读取文件名为 `filename` 的文件，并将该文件的内容显示出来



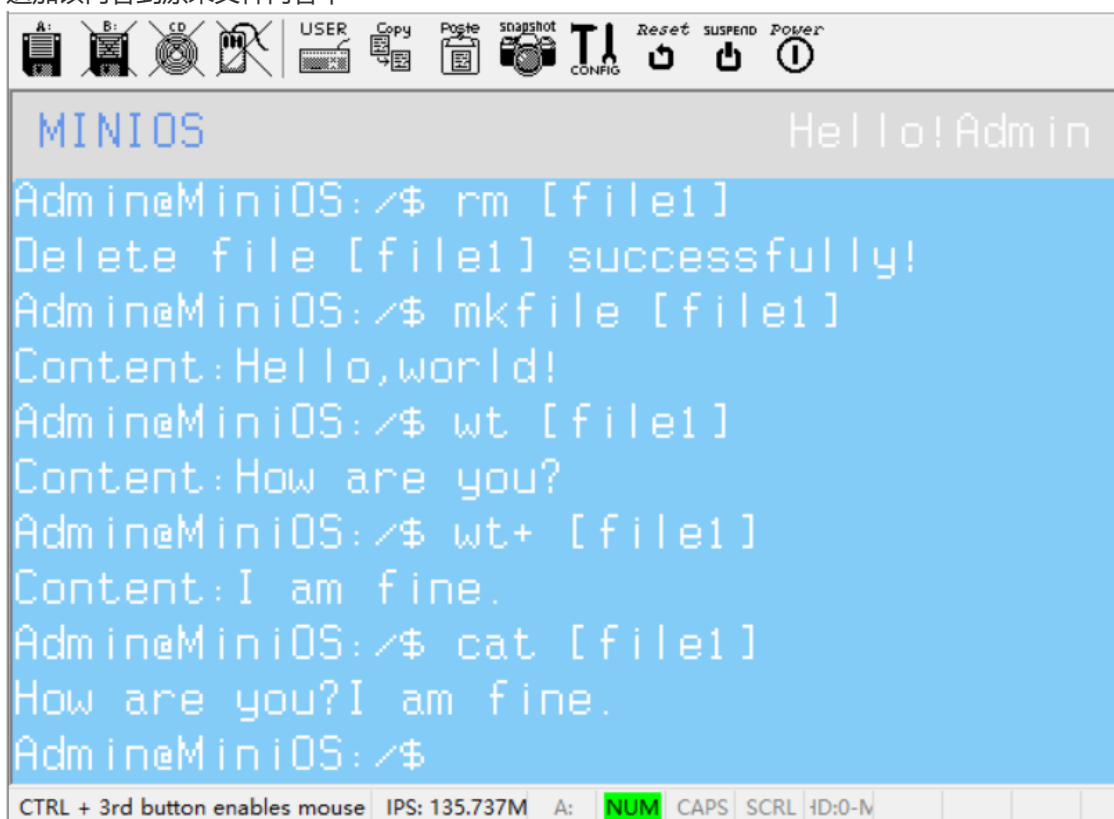
- 删除文件：输入 `rm [filename]` 指令删除文件名为 `filename` 的文件，同时将其从文件目录中删除



- 写文件覆盖内容：输入 `wt [filename]` 指令编辑文件名为 `filename` 的文件，同时输入新的内容将原来文件的内容覆盖

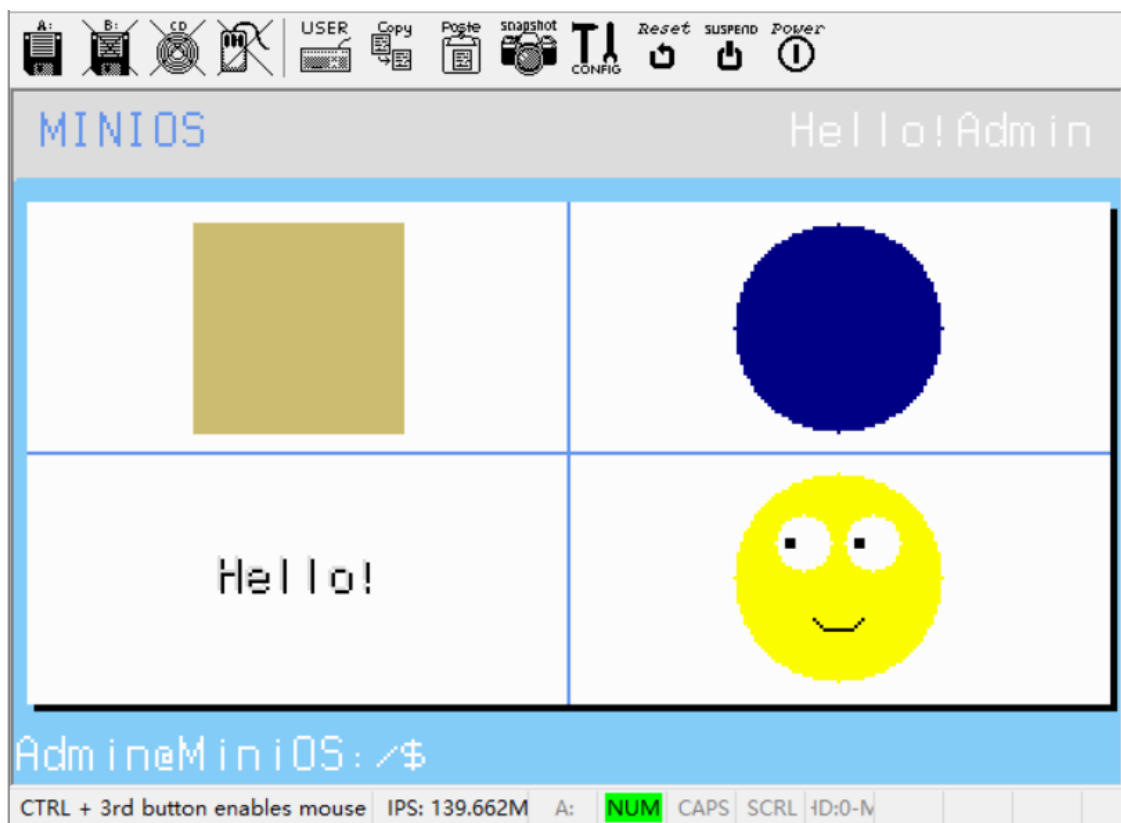


- 写文件追加内容：输入 `wt+ [filename]` 指令编辑文件名为 `filename` 的文件，同时输入新的内容追加该内容到原来文件内容中

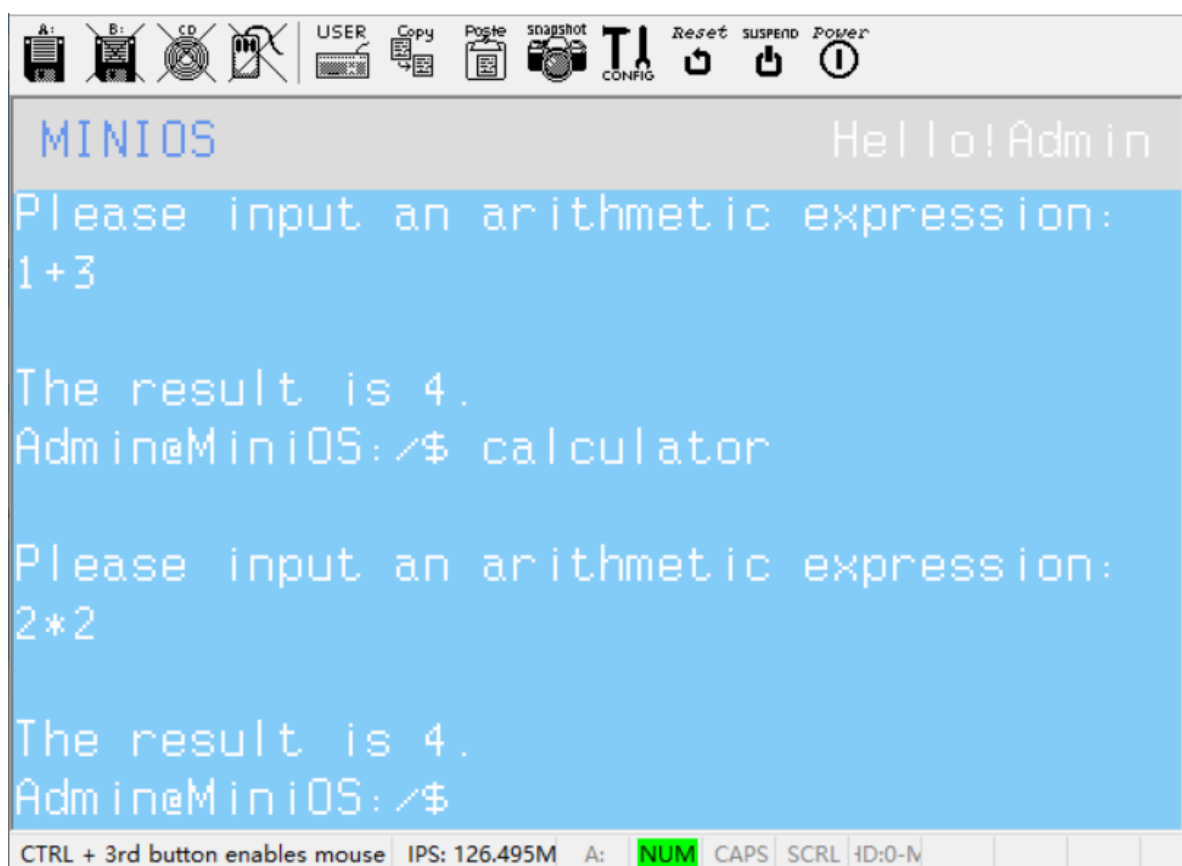


3.6 用户级应用——画图

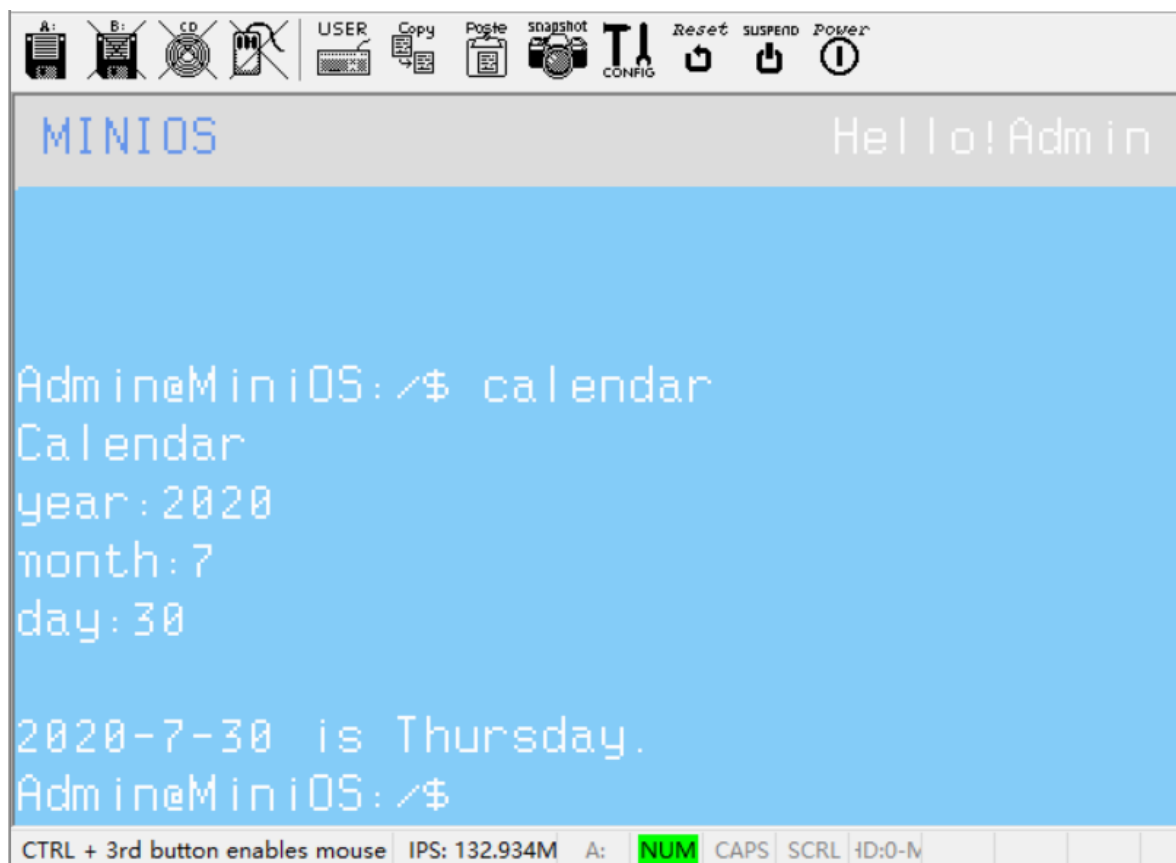
- 输入 `draw` 指令，执行画图程序，该程序基于图形管理模块提供的画图接口，能够绘制像素点、文字、直线、矩形、圆形



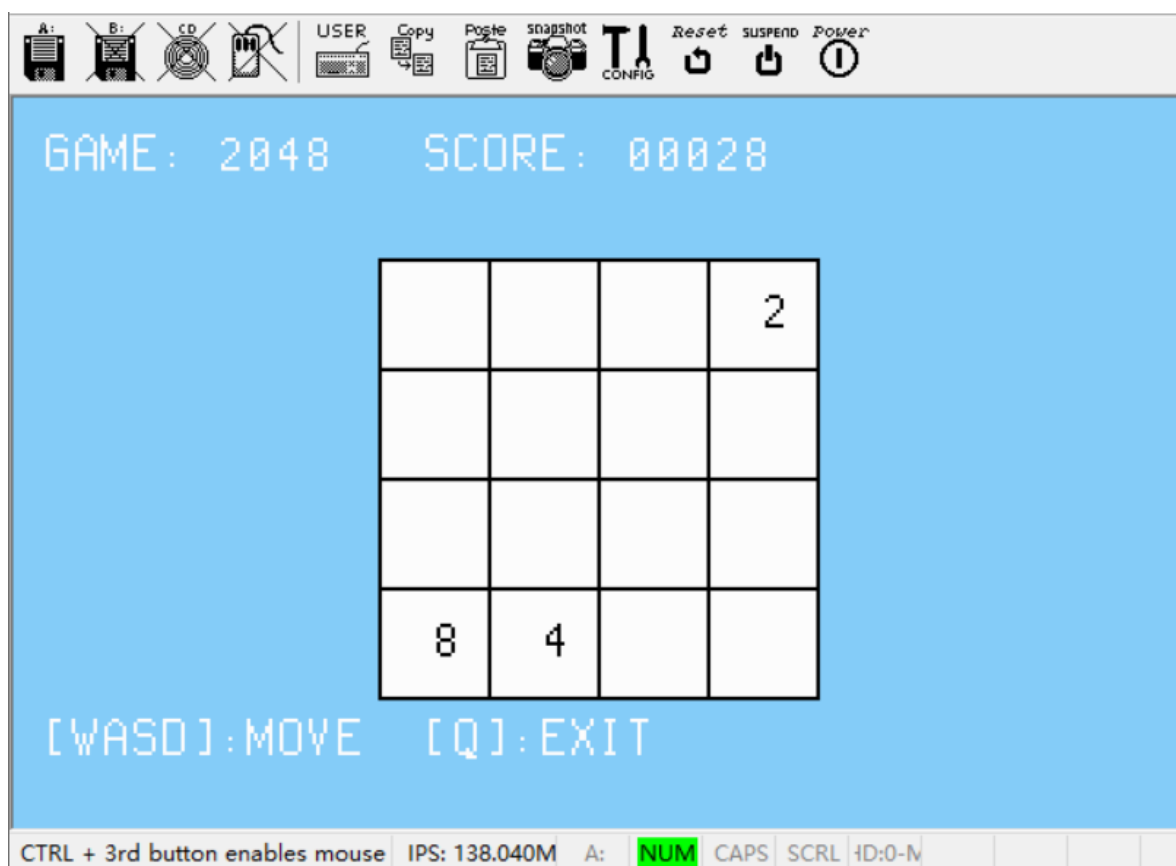
3.7 用户级应用——计算器



3.8 用户级应用——日历



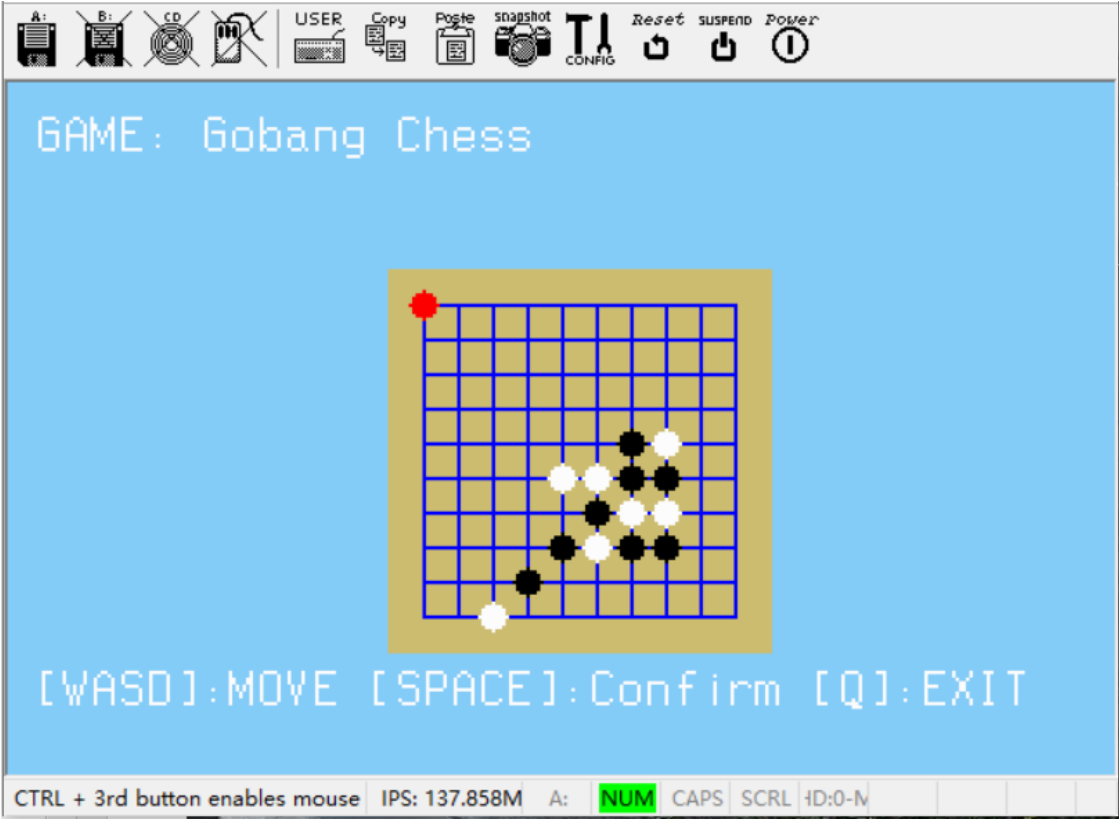
3.9 用户级应用——2048



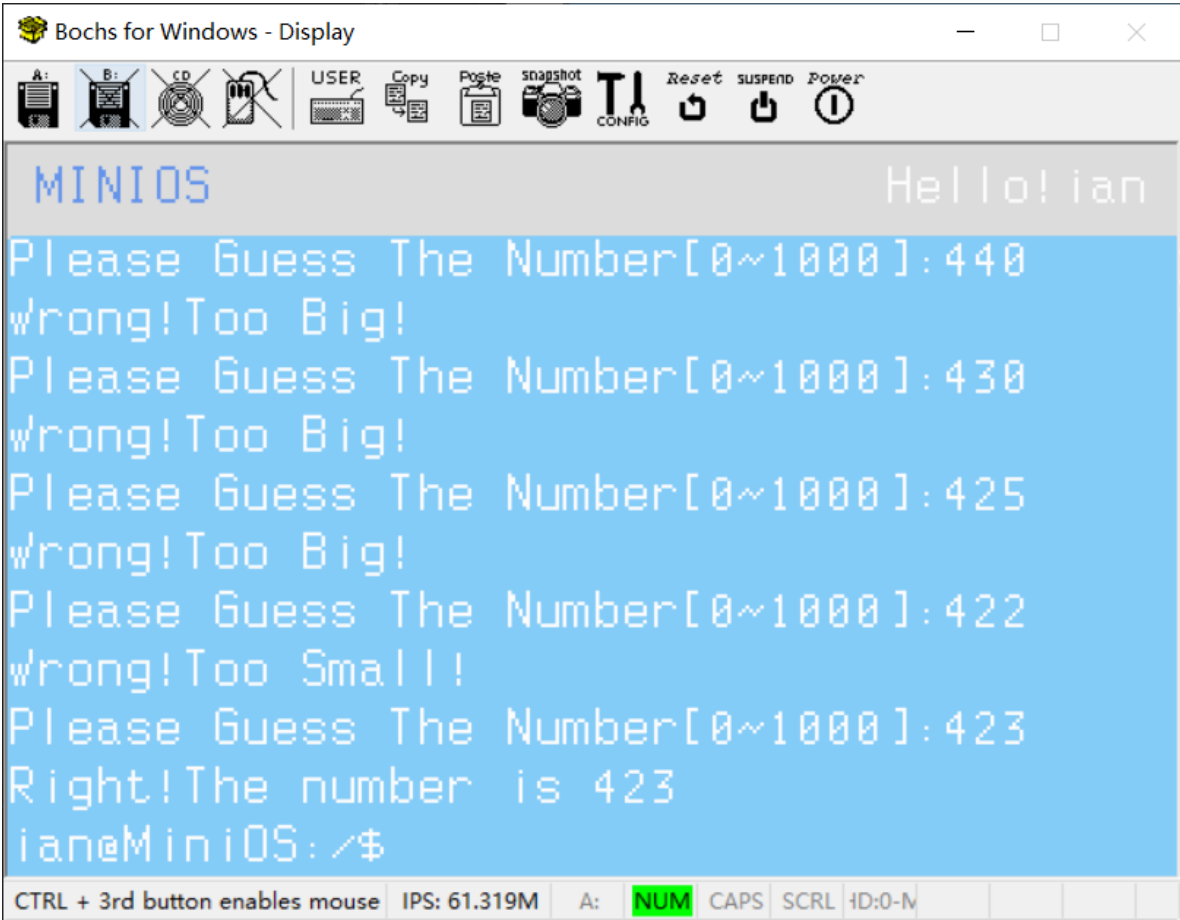
3.10 用户级应用——五子棋

- **游戏规则：** 调用指令 `chess` 启动五子棋游戏后，可以看到五子棋的游戏界面。玩家的游戏棋子为白子，电脑AI的游戏棋子为黑子。玩家通过方向键 `W`、`A`、`S`、`D` 控制白棋在界面进行移动，使用按键 `SPACE` 进行落子确认，与电脑ai进行五子棋对战。当有一方的同一颜色的棋子在横行竖行

斜行连成五个子，游戏结束。可以通过按键 `q\Q` 退出游戏

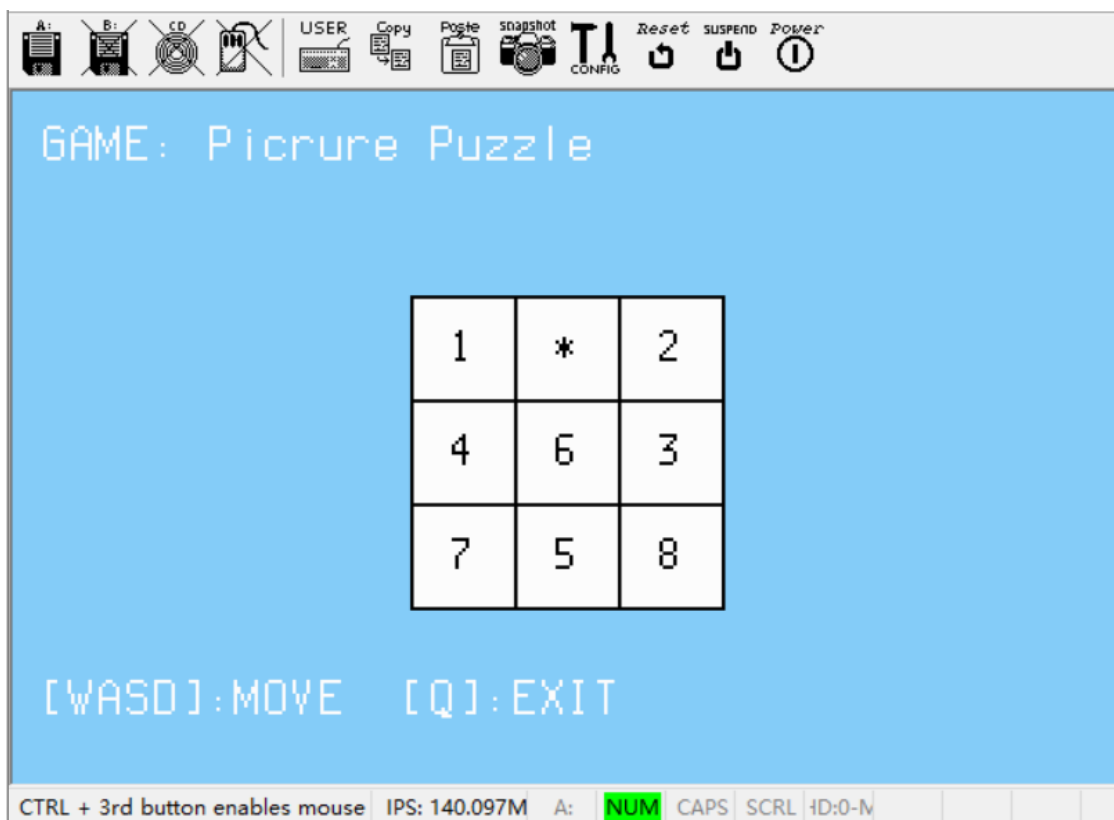


3.11 用户级应用——猜数字



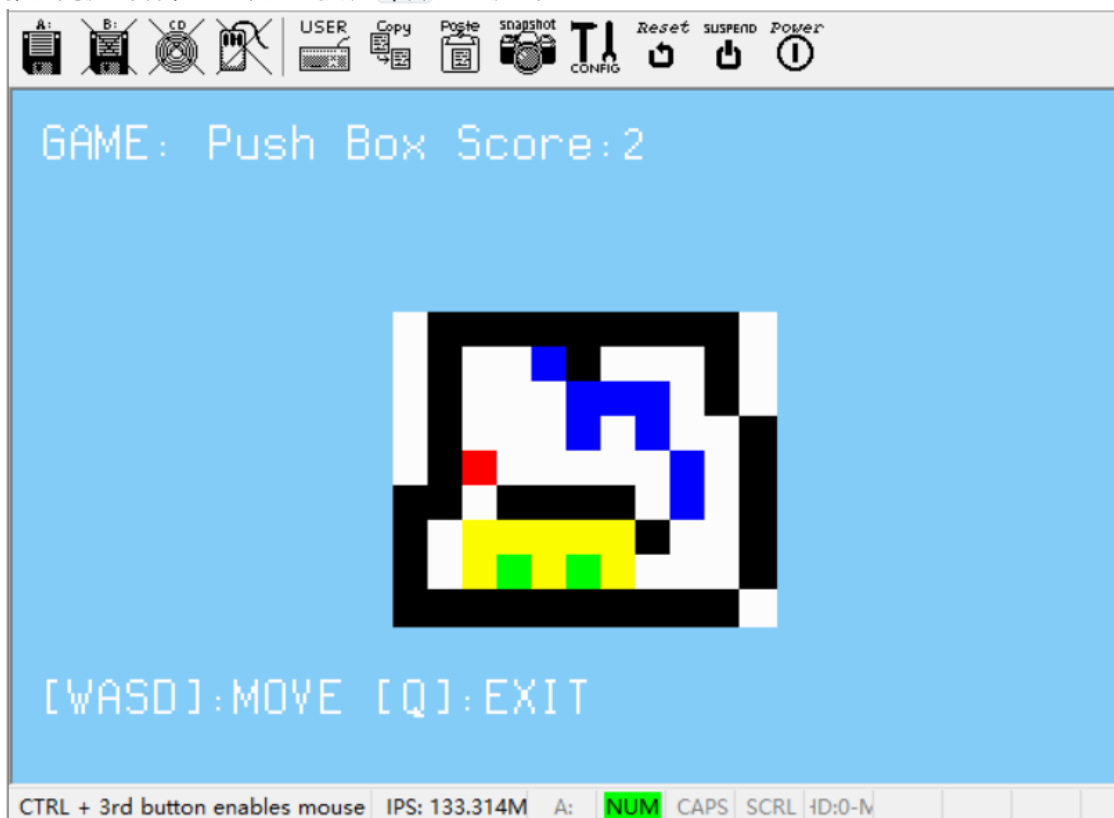
3.12 用户级应用——拼图

- **游戏规则：** * 是被控制的位置， `awsd` 控制 * 的移动。当所有数字顺序排放并且 * 位于右下角时，游戏结束。



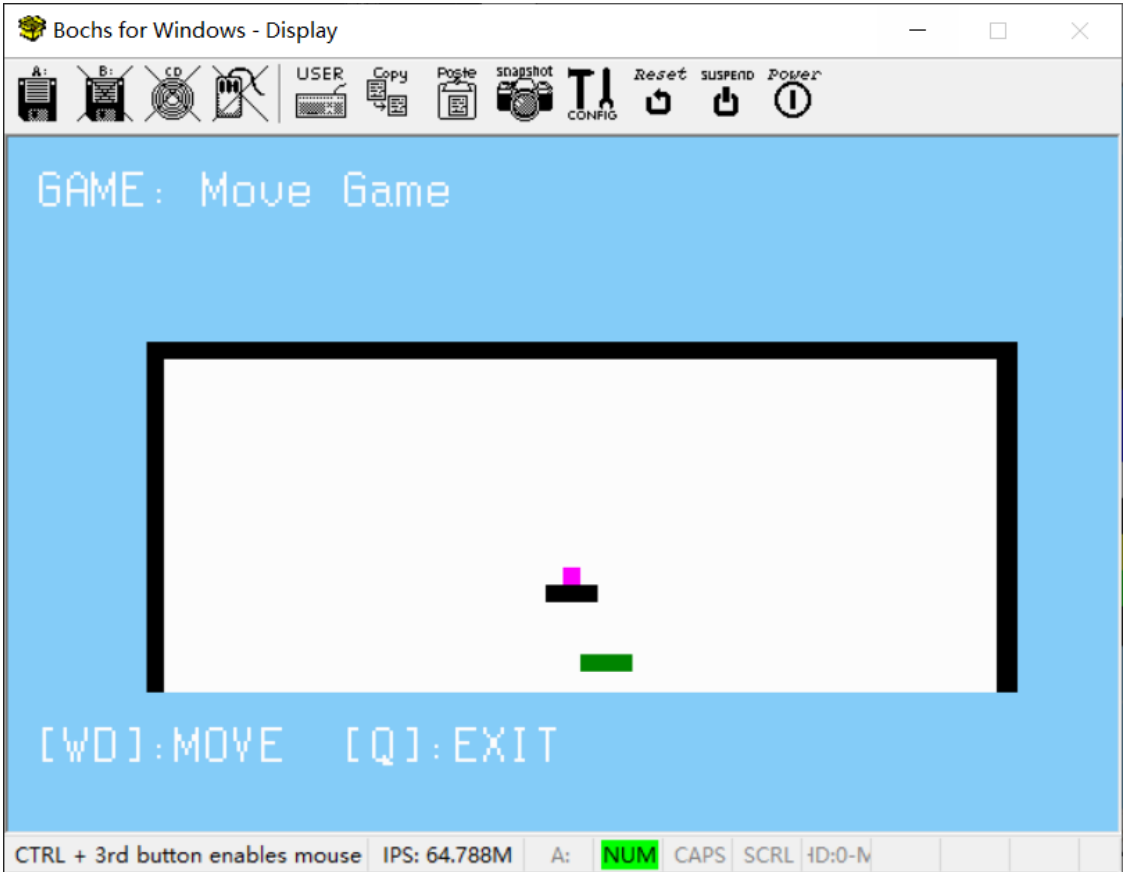
3.13 用户级应用——推箱子

- 游戏规则：** 调用指令 `pushbox` 启动推箱子游戏后，可以看到推箱子的游戏界面。红色方块为玩家控制的人物，蓝色方块为游戏中等待推动的箱子，黄色方块为游戏中箱子推送的目的地，绿色方块为成功推送到目的地后的箱子，紫色方块为进入目的地的玩家控制的人，黑色方块为游戏中的墙壁，白色方块为游戏中可以移动的空白区域。玩家通过方向键 `W`、`A`、`S`、`D` 控制游戏人物进行上下左右的移动推动箱子进入目的地。游戏界面的分数显示成功推进目的地的箱子个数，箱子全部推进则游戏结束。可以通过按键 `q\Q` 退出游戏

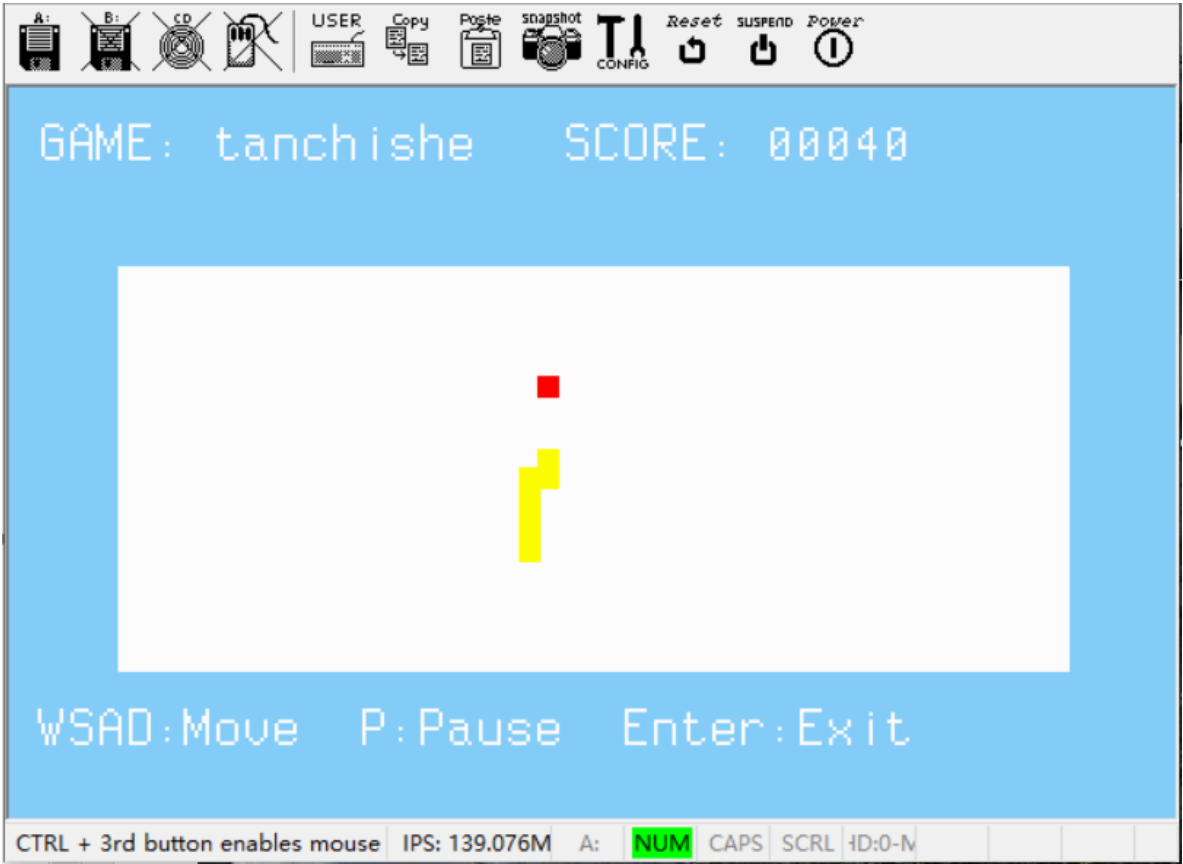


3.14 用户级应用——MoveGame

- **游戏规则：** w和d 控制小球的运动，红色条踩了会立即消失，蓝色条加速向左，绿色条加速向右，黑色条可以正常踩。玩家每移动一次小球，游戏运行一帧。当小球落到最底部时游戏结束。



3.15 用户级应用——贪吃蛇



4.功能实现

4.1 图形界面

- 图形管理模块(graphics/main.c):

```
PUBLIC void task_graphics()
{
    printf("{GRAPHICS} Task GRAPHICS begins.\n");

    while (1) {
        send_recv(RECEIVE, ANY, &graphics_msg);
        int src = graphics_msg.source;
        int reply = 1;

        int msgtype = graphics_msg.type;

        switch (msgtype) {
            case DRAW:
                graphics_msg.RETVAL=do_drawBMP();
                break;
            case TOPBAR:
                graphics_msg.RETVAL=do_topBar();
                break;
            case CLEAR:
                do_clear();
                break;
            case PIXEL:
                graphics_msg.RETVAL=do_drawPixel();
                break;
            case FONT:
                do_drawFont();
                break;
            case BOX:
                do_drawBox();
                break;
            case CIRCLE:
                do_drawCircle();
                break;
            case LINE:
                do_drawLine();
                break;
        }
    }
}
```

```

        case MODE:
            do_changeColorMode();
            break;
        default:
            dump_msg("GRAPHICS::unknown msg", &graphics_msg);
            assert(0);
            break;
    }

    if (reply) {
        graphics_msg.type = SYSCALL_RET;
        send_recv(SEND, src, &graphics_msg);
    }
}
}

```

- 绘图函数(lib/graphics.c):

```

/*****
 *
 *                               clearAll
 *****/
PUBLIC void clearAll()
{
    MESSAGE msg;
    msg.type = CLEAR;
    send_recv(BOTH, TASK_GRAPHICS, &msg);
}

/*****
 *
 *                               changeColorMode
 *****/
PUBLIC void changeColorMode()
{
    MESSAGE msg;
    msg.type = MODE;
    send_recv(BOTH, TASK_GRAPHICS, &msg);
}

/*****
 *
 *                               drawFont
 *****/
PUBLIC void drawFont(int x, int y, char *s, char c)
{
    MESSAGE msg;
    msg.type = FONT;
    msg.OFFSETX=x;
    msg.OFFSETY=y;
    msg.COLOR3=c;
    msg.STRING = (void*)s;
    msg.STRLEN = strlen(s);

    send_recv(BOTH, TASK_GRAPHICS, &msg);
}

```



```

/*****
 *
 * drawPixel
 *****/
PUBLIC int drawPixel( int x, int y,unsigned char c)
{
    MESSAGE msg;
    msg.type = PIXEL;
    msg.M1I1=c;
    msg.M1I2=x;
    msg.M1I3=y;
    send_recv(BOTH, TASK_GRAPHICS, &msg);
    return msg.RETVAL;
}

/*****
 *
 * drawBox
 *****/
PUBLIC void drawBox( int x0, int y0, int x1, int y1,unsigned char c)
{
    MESSAGE msg;
    msg.type = BOX;
    msg.COLOR4=c;
    msg.X1=x0;
    msg.Y1=y0;
    msg.X2=x1;
    msg.Y2=y1;
    send_recv(BOTH, TASK_GRAPHICS, &msg);
}

/*****
 *
 * drawLine
 *****/
PUBLIC void drawLine(int x0, int y0, int x1, int y1,unsigned char c)
{
    MESSAGE msg;
    msg.type = LINE;
    msg.COLOR4=c;
    msg.X1=x0;
    msg.Y1=y0;
    msg.X2=x1;
    msg.Y2=y1;
    send_recv(BOTH, TASK_GRAPHICS, &msg);
}

/*****
 *
 * drawCircle
 *****/
PUBLIC void drawCircle( int x, int y,int radius,unsigned char c)
{
    MESSAGE msg;
    msg.type = CIRCLE;
    msg.M1I1=c;
    msg.M1I2=x;
    msg.M1I3=y;
    msg.M1I4=radius;
    send_recv(BOTH, TASK_GRAPHICS, &msg);
}

```

- 显示BMP图片(graphics/main.c graphics/graphics.c):

```
int do_drawBMP()
{
    char pathname[MAX_PATH];

    int name_len = graphics_msg.NAME_LEN; /* length of filename */
    int src = graphics_msg.source; /* caller proc nr. */
    assert(name_len < MAX_PATH);
    phys_copy((void*)va2la(TASK_GRAPHICS, pathname),
              (void*)va2la(src, graphics_msg.PATHNAME),
              name_len);
    pathname[name_len] = 0;

    return readBMP(pathname);
}
```

```

PUBLIC int readBMP(char* filename)
{
    char buff[512];
    char *p=VGA_MEM_BASE;
    int fd = open(filename, O_RDWR),n,j,i,buffSize=512;
    if(fd==-1)
    {
        return 0;
    }
    for(i=0;i<BMP_HEAD_SIZE/buffSize;i++)
    {
        read(fd,buff,buffSize);

        read(fd,buff,BMP_HEAD_SIZE%buffSize);

        for(i=0;i<SCREEN_SIZE/buffSize;i++)
        {
            n=read(fd,buff,buffSize);
            for(j=0;j<n;j++)
            {
                buff[j]%=16;
                if(buff[j]<0)    buff[j]+=16;
                p[j]=buff[j];
            }
            p+=n;
        }

        n=read(fd,buff,SCREEN_SIZE%buffSize);
        for(j=0;j<n;j++)
            p[j]=buff[j];
        close(fd);
        return 1;
    }
}

```

4.2 系统级应用——多用户登录

- kernel/tools/user.c:

```
//用户数据
struct Users users[MAX_USER_COUNT];
char currentUser[128] = "/";
int usercount = 0;
int currentUserIndex=-1;

int verify() ...

void initUsers() ...

//根据用户名查找用户下标
int searchUser(char *username) ...
//登录
void login() ...

//显示所有用户
void showUsers() ...

//添加用户
void addUser(char *username, char *password) ...

//删除用户
void delUser(char *username) ...

//将用户数据保存到文件
void updateUsers() ...
void passwd(char *username) ...
//修改用户密码
void modifyPassword(int index) ...
//获取管理员权限
void sudo() ...
```

4.3 系统级应用——进程管理

- kernel/tools/process.c:

```
//显示进程
void showProcess() ...

//停止进程
void killpro(char *a) ...

//暂停进程
void pausepro(char *a) ...

//恢复进程
void resume(char *a) ...
```

4.4 系统级应用——多功能控制台

- kernel/tty.c:

```
/*清空当前输入命令*/
PRIVATE void empty(TTY* tty) ...

/*替换当前输入命令*/
PRIVATE void putstring(TTY *tty, const char* str) ...

/*将输入的命令计入历史命令中*/
PRIVATE void remember(TTY *tty) ...

/*匹配命令*/
PRIVATE void match(TTY* tty) ...
```

- kernel/console.c

```
void putfont(int x,int y,char ch)
{
    x=x*FONT_WIDTH;
    y=y*FONT_HEIGHT+TOPBAR_HEIGHT;
    boxfill8(SCREEN_WIDTH,DEFAULT_BACKGROUND,x,y,x+FONT_WIDTH,y+FONT_HEIGHT);
    putfonts8_char( x,y,DEFAULT_FRONT, ch);
}
```

```
PRIVATE void w_copy(CONSOLE *con,unsigned int dst,unsigned int src)
{
    int k,size=SCR_SIZE-SCR_WIDTH;
    char *cbuf=con->cbuf;
    for(k=0;k<size;k++)
    {
        cbuf[dst+k]=cbuf[src+k];
    }
    for(k=size;k<SCR_SIZE;k++)
    {
        cbuf[dst+k]=' ';
    }
    for(k=0;k<SCR_SIZE;k++)
    {
        int x=k%SCR_WIDTH;
        int y=k/SCR_WIDTH;
        putfont(x,y,cbuf[k]);
    }
}
```

- kernel/main.c/shabby_shell:

```
//执行命令
if (strcmp(argv[0], "clear") == 0) ...
else if (strcmp(argv[0], "users") == 0) ...
else if (strcmp(argv[0], "useradd") == 0) ...
else if (strcmp(argv[0], "userdel") == 0) ...
else if (strcmp(argv[0], "passwd") == 0) ...
else if (strcmp(argv[0], "ls") == 0) ...
else if (strcmp(argv[0], "mkfile") == 0) ...
else if (strcmp(argv[0], "cat") == 0) ...
else if (strcmp(argv[0], "rm") == 0) ...
else if (strcmp(argv[0], "wt") == 0) ...
else if (strcmp(argv[0], "wt+") == 0) ...
else if (strcmp(argv[0], "logout") == 0) ...
else if (strcmp(argv[0], "sudo") == 0) ...
else if (strcmp(argv[0], "colormode") == 0) ...
else if (strcmp(argv[0], "bmp") == 0) ...
else if (strcmp(argv[0], "2048") == 0) ...
else if (strcmp(argv[0], "tanchishe") == 0) ...
else if (strcmp(argv[0], "calendar") == 0) ...
else if (strcmp(argv[0], "calculator") == 0) ...
else if (strcmp(argv[0], "pintu") == 0) ...
else if (strcmp(argv[0], "chess") == 0) ...
else if (strcmp(argv[0], "pushbox") == 0) ...
else if (strcmp(argv[0], "movegame") == 0) ...
else if (strcmp(argv[0], "guessnum") == 0) ...
else if (strcmp(argv[0], "proc") == 0) ...
else if (strcmp(argv[0], "kill") == 0) ...
else if (strcmp(argv[0], "pause") == 0) ...
else if (strcmp(argv[0], "resume") == 0) ...
else if (strcmp(argv[0], "help") == 0) ...
else
{
    if(argc>0)
        printf("Command '%s' not found\n",argv[0]);
    continue;
}
```

4.5 系统级应用——文件管理

- kernel/tools/fs.c:

```
//文件数据
char currentFolder[128] = "/";
char filepath[128] = "";
char files[20][128];
int fileCount=0;

//覆盖文件
void editCover(char *filepath, char *buf) ...

//追加文件
void editAppand(char *filepath, char *buf) ...

/* ls */
void ls() ...

void pathFilter(char* bufr) ...

//创建文件路径
void createFilepath(char* filename) ...

/* 创建文件*/
void createFile(char *file) ...

/* 读取文件*/
void readFile(char *file) ...

//删除文件
void delFile(char *file) ...
//覆盖文件
void ModifyFile(char *file) ...
//追加文件
void AppendFile(char *file) ...

//清空数组
void clearArr(char *arr, int length) ...
// 读取用户文件列表
void readUserFiles() ...
// 更新用户文件列表
void updateUserFiles() ...
```


4.6 用户级应用——画图

- command/draw.c:

```
int centerX=SCREEN_WIDTH/2,centerY=PANELSIZE/2+TOPBAR_HEIGHT;
int sizeX=SCREEN_WIDTH/2-FONT_WIDTH/2;
int sizeY=PANELSIZE/2-FONT_HEIGHT/2;

int main(int argc, char * argv[])
{
    clearAll();
    drawPanel();
    drawGrid();
    boxTest();
    circleTest();
    fontTest();
    smile();
    return 0;
}

/*绘制画布*/
void drawPanel() ...
/*绘制网格*/
void drawGrid() ...
/*绘制长方体*/
void boxTest() ...
/*绘制圆*/
void circleTest() ...
/*绘制文字*/
void fontTest() ...
/*绘制笑脸*/
void smile() ...

void delay(int time)...
```

4.7 用户级应用——计算器

- kernel/tools/calculator.c:

```
char str[400] = "\0";
bool flag = true;

int num_stack[200] = { 0 };
int num_stack_top = -1;
int left = 0, right = 0;

bool num_stack_is_empty(){ ...

void num_stack_push(int num) { ...

int num_stack_pop() { ...

void num_stack_clear() { ...

int num_stack_get_top() { ...

char op_stack[200] = "\0";
int op_stack_top = -1;

bool op_stack_is_empty() { ...

void op_stack_push(char ch) { ...

char op_stack_pop() { ...

char op_stack_get_top() { ...
```

```

void op_stack_clear() { ...

bool is_digit(char ch) ...

int isp(char ch){ ...

int icp(char ch){ ...
int pow(int a,int b) ...

bool get_two_num() { ...

void do_operator(char ch) { ...

void run() { ...

bool format() { ...

void runcalculator() {
    printf("\nPlease input an arithmetic expression:\n");
    int r = read(0, str, 400);
    str[r] = 0;
    if (!format()) {
        printf("\nThe arithmetic expression you input is not correct.Please enter again.\n");
        return;
    }
    else{
        run();
        if (flag) {
            int number = num_stack_get_top();
            printf("\nThe result is %d.\n", number);
        }
    }
}
}

```

4.8 用户级应用——日历

- kernel/tools/calendar.c:

```
void runcalendar() {
    int year, month, day;
    printf("Calendar\n");
    printf("year:");
    year=readint();
    printf("month:");
    month=readint();
    printf("day:");
    day=readint();
    // printf("%d\t%d\t%d\n",year,month,day);
    print_weekday(year, month, day);
}
```

```
bool is_leap(int year) { ...
```

```
int day_count(int year, int month) { ...
```

```
int total_day(int year,int month,int day) { ...
```

```
int weekday(int year, int month, int day){ ...
```

```
void print_weekday(int year, int month, int day) { ...
```

```
void print_month(int year,int month) { ...
```

```
void print_total_day(int year, int month, int day) { ...
```

4.9 用户级应用——2048

- kernel/gmae/2048.c:

```
void run2048()
{
    init_game();
    loop_game();
}

void loop_game() { ...

void reset_game() { ...

void add_rand_num() { ...

int get_null_count() { ...

void check_game_over() { ...

void move_left() { ...

void move_right() { ...

void move_up() { ...

void move_down() { ...

void refresh_screen() { ...

void init_game() { ...

void release_game() { ...
```

4.10 用户级应用——五子棋

- kernel/gmae/chess.c:

```
int chess[10][10];          /* 10*10的棋盘 */
int a, b;                   /* a b为玩家下棋坐标*/
int c, d, x;                /* c d为电脑下棋坐标, x为剩余空位置*/
int currentX=0,currentY=0;

int isStart = 0;
int isPainting = 0;
int state = 1;              /* 1为无效, 0为有效*/
static const int centerX=SCREEN_WIDTH/2;
static const int centerY=PANELSIZE/2+TOPBAR_HEIGHT;
static int gridSizeX=10;
static int gridSizeY=10;
static int chessSize=4;

// 函数声明
PRIVATE void start();       /* 程序开始 */
PRIVATE void draw();        /* 棋盘绘制 */
PRIVATE int win(int p,int q); /* 由棋子判断胜利情况 */
PRIVATE void AI(int *p,int *q); /* 电脑下棋返回p q棋子坐标 */
PRIVATE int value(int p,int q); /* 估值函数 */

PRIVATE int formationStrategy(int n, int p, int q); /* 棋型号判断 */
PRIVATE void moveStep(int n, int *x, int *y); /* 棋盘棋子移动 */

PRIVATE void start() ...

PRIVATE void draw() ...

PRIVATE int win(int p, int q) ...

PRIVATE void AI(int *p, int *q) ...

// 估值函数, 由井字棋规则给出
PRIVATE int value(int p,int q) ...

// 由井字棋规则定义棋型号, 判断棋子周围棋子对应棋形号并返回
// 棋型号注解: 己活000-003, 己冲010-013, 对活100-103, 对冲110-113, 己空活020-0
PRIVATE int formationStrategy(int n, int p, int q) ...

PRIVATE void moveStep(int n, int *x, int *y) ...

void runchess() ...
```

4.11 用户级应用——猜数字

- kernel/gmae/guessnum.c:

```
void runguess()
{
    int n=rand()%1000,g,count=0;;
    while (true)
    {
        if(count>=10)
        {
            printf("Loss!You have guess %d times!\n",count);
            return;
        }
        printf("Please Guess The Number[0~1000]:");
        g=readint();
        if(g==n)
        {
            printf("Right!The number is %d\n",n);
            return;
        }
        else if (g<n)
        {
            printf("Wrong!Too Small!\n");
        }
        else
        {
            printf("Wrong!Too Big!\n");
        }
        count++;
    }
}
```

4.12 用户级应用——拼图

- kernel/gmae/pintu.c:

```
PRIVATE void initScreen() ...

PRIVATE bool judge(int x,int y) ...

PRIVATE void init_board() ...

PRIVATE bool sucess() ...

PRIVATE void print_board() ...

PRIVATE void work() ...

void runpintu()
{
    while(1)
    {
        initScreen();
        work();
        char *s="TRY THE GAME AGAIN? [Y/N]";
        drawFont(centerX-strlen(s)/2*FONT_WIDTH,centerY+FONT_HEIGHT,s,COLOR_BRIGHTYELLOW);

        u32 key=kbhit();
        if(!(key=='y' || key=='Y'))
            return;
    }
}
```

4.13 用户级应用——推箱子

- kernel/gmae/pushbox.c:

```
// 推箱子
int pi=0;
int pj=0;
int count;
static const int centerX=SCREEN_WIDTH/2;
static const int centerY=PANELSIZE/2+TOPBAR_HEIGHT;
static int gridSize=10;
static int gridSizeY=10;

void draw_map(int map[9][11]) ...

void boxMenu() ...

void runpushbox() ...
```

4.14 用户级应用——MoveGame

- kernel/gmae/smggame.c:

```
#define go(x,y) swap(&map[nx][ny],&map[nx+x][ny+y]),nx=nx+x,ny=ny+y

static const int centerX=SCREEN_WIDTH/2;
static const int centerY=PANELSIZE/2+TOPBAR_HEIGHT;
static int gridSizeX=5;
static int gridSizeY=5;

char map[100][100];
int maxscore;
int nx,ny;
int speed,score,downtime;
bool difficult=0;
bool tool=0;
bool died=0;
int nowt;
int st;
int _a,_b;
int o=0;//用于隔行生成'###'
PRIVATE bool ifdied(){...}
PRIVATE void swap(char* a,char* b){...}
PRIVATE void up(){...}
PRIVATE void init(){...}
PRIVATE void down(){...}
int na,nb;
PRIVATE void nowt(){...}

PRIVATE void print(){...}
PRIVATE void game(){...}

PRIVATE void initScreen(){...}
void runsmgame(){
    initScreen();
    game();
    printf("Final Score:%d\n",score);
}
```

4.15 用户级应用——贪吃蛇

- kernel/gmae/tanchishe.c:

```
static bool if_game_over;
static bool if_exist_game = false;
static bool stop = false;
static const int width = 50;
static const int height = 20;
static int x, y, food_x, food_y, score;
static int tail_x[200], tail_y[200];
static int ntail = 3;
static const int drawSize=5;
static const int centerX=SCREEN_WIDTH/2;
static const int centerY=PANELSIZE/2+TOPBAR_HEIGHT;

static void menu();
static void setup();
static void draw();
static void input();
static void loop_game();

typedef enum{
    STOP = 0,
    DIR_LEFT,
    DIR_RIGHT,
    DIR_UP,
    DIR_DOWN
}Direction;
Direction Dir;
```

```
void menu(){ ...  
  
void setup(){ ...  
  
void draw(){ ...  
  
void input() ...  
  
void logic() ...  
void loop_game() { ...  
  
void runtanchishe()  
{  
    menu();  
    setup();  
    draw();  
    loop_game();  
    clearAll();  
}
```
