**操作系统项目文档**

学号：1352996

姓名：李秉泽

邮箱：[741090870@qq.com](mailto:741090870@qq.com)

github用户名: breakfast521

**一 引言**

此项目作为操作系统课程的课程设计，通过完成项目来加深对操作系统的理解，提高编程能力。

我们组是3人小组，但3个人独立开发的自己的用户应用，撰写文档。

**二 开发工具与环境**

VMware虚拟机，Bochs，Ubuntu （之后换为windows ，QEMU）

**三 参考资料**

《Orange’s一个操作系统的实现》-于渊

《30天自制操作系统》-川合秀实

**四 项目需求**

在参考代码实现的操作系统上，通过调用其API，实现一个用户应用程序。

**五 项目简介**

我所完成的是名为plane的一个躲避障碍物小游戏。模拟出一架飞机（^$^）,通过↑↓←→四个方向键控制飞机的移动，游戏窗口的上方会不断出现陨石（@），玩家通过控制飞机躲避陨石，随着存活时间增加，“陨石”的下落速度会越来越快。“飞机”接触到“陨石”则游戏结束，提示“GAME OVER”。

**六 代码说明**

此次项目是利用川合秀实《30天自制操作系统》一书中所给的参考代码实现的操作系统所写的，需要调用系统提供的接口。

程序中有一个名为Enemy的结构体，表示飞机的敌人“陨石”

struct Enemy{

int x;

int y;

}enemy[8];

只有两个属性，x和y表示位置，以像素点表示。

主函数中首先声明了一些要用到的变量。然后通过api\_openwin生成一个窗口。再通过api\_boxfilwin将窗口中间填充为黑色。

void HariMain(void)

{

int win, i;

char buf[200 \* 200];

char keyflag[4];

win = api\_openwin(buf, 200, 200, -1, "test");

api\_boxfilwin(win, 5, 25, 195, 195, 0);

……………..

主函数中有程序的主循环

for (;;) {

int e=0;

if(flag == 0)

{

for(e = 0;e < 8; e++)

{

enemy[e].y = rand()/1310+25;

enemy[e].x = rand()/173 + 3;

}

flag = 1; //只生成一次

}

for(e = 0; e<8;e++)

{

api\_putstrwin(win,enemy[e].x, enemy[e].y, 0, 1, "@"); //用黑色覆盖轨迹

enemy[e].y = enemy[e].y+5; //move down

api\_putstrwin(win,enemy[e].x, enemy[e].y, 3, 1, "@"); //显示新位置

if(enemy[e].y > 180){

api\_putstrwin(win,enemy[e].x, enemy[e].y, 0, 1, "@");

enemy[e].y = rand()/1310+25;

enemy[e].x = rand()/173 + 3;

if(s>99999)

s = s - 99999 ; //加速

}

if(planex >enemy[e].x-20 && planex <enemy[e].x+3) //判断碰撞

if(planey >enemy[e].y-3 && planey <enemy[e].y+3)

{

api\_putstrwin(win , 100, 100, 1, 9, "GAME OVER");

while(1)

{

if (api\_getkey(1) == 0x0a)

{

goto start;

} /\* 重新开始 \*/

}

}

}

int t = 0;

for(t=0; t<s; t++){} // wait.....

move(win,1, timer, keyflag);

}

通过flag全局变量在第一次执行循环时，生成8个enemy，在生成时利用随机数将他们的横纵左边分散。rand（）会得到0-32768的一个随机数，通过一些计算将纵坐标分散的25-50，横坐标分散到3-195.

enemy[e].y = rand()/1310+25;

enemy[e].x = rand()/173 + 3;

主循环中还有一个循环，用来使生成的enemy不断下移，每次下移5个像素。下移之前要把原来的位置用黑色覆盖掉，否则会留下轨迹。用打印函数改变颜色即可实现。

api\_putstrwin(win,enemy[e].x, enemy[e].y, 0, 1, "@"); //用黑色覆盖轨迹

enemy[e].y = enemy[e].y+5; //move down

api\_putstrwin(win,enemy[e].x, enemy[e].y, 3, 1, "@"); //显示新位置

移动后判断enemy的位置，如果他们的纵坐标超过了180，则利用随机数重置他们的位置，和初始生成的位置的方法相同。重置位置后代表成功躲过这个障碍物，所有陨石加速。通过S递减99999来缩减空循环时间，从而加速。

if(enemy[e].y > 180){

api\_putstrwin(win,enemy[e].x, enemy[e].y, 0, 1, "@");

enemy[e].y = rand()/1310+25;

enemy[e].x = rand()/173 + 3;

if(s>99999)

s = s - 99999 ; //加速

}

另外一个模块就是判断碰撞。

if(planex >enemy[e].x-20 && planex <enemy[e].x+3) //判断碰撞

if(planey >enemy[e].y-3 && planey <enemy[e].y+3)

{

api\_putstrwin(win , 100, 100, 1, 9, "GAME OVER");

while(1)

{

if (api\_getkey(1) == 0x0a)

{

goto start;

} /\* enter重新开始 \*/

}

}

在循环中检查“飞机”的横纵坐标和enenmy的横纵坐标，如果飞机的位置在enemy位置的一定范围内，则判定为碰撞，游戏结束，提示“GAME OVER”，提示用到了另一个系统的API。api\_putstrwin(win , 100, 100, 1, 9, "GAME OVER");

在主循环中还有一个空循环。用来控制enemy的下落速度。

int t = 0;

for(t=0; t<s; t++){} // wait.....

除了主函数，还是一个重要的move函数，用来实现监听键盘的方向键控制飞机移动，在移动的同时也要用黑色覆盖原来的位置，清除轨迹。

void move(int win ,int i, int timer, char \*keyflag)

{

int j;

if (i > 0) {

api\_settimer(timer, i);

i = 128;

} else {

i = 0x0a; /\* Enter \*/

}

for (;;) {

j = api\_getkey(1);

if (i == j) {

break;

}

if (j == '4') {

keyflag[0 /\* left \*/] = 1;

if (planex > 5) {api\_putstrwin(win, planex, planey, 0, 3, "^$^"); planex -= 5; } //清除轨迹

}

if (j == '6') {

keyflag[1 /\* right \*/] = 1;

if (planex <= 185) {api\_putstrwin(win, planex, planey, 0, 3, "^$^"); planex += 5; }

}

if (j == '8') {

keyflag[2 /\* up \*/] = 1;

if (planey >= 25) {api\_putstrwin(win, planex, planey, 0, 3, "^$^"); planey -= 5; }

}

if (j == '2') {

keyflag[3 /\* down \*/] = 1;

if ( planey <= 180) {api\_putstrwin(win, planex, planey, 0, 3, "^$^"); planey += 5; }

}

}

api\_putstrwin(win, planex, planey, 3, 3, "^$^");

return;

}

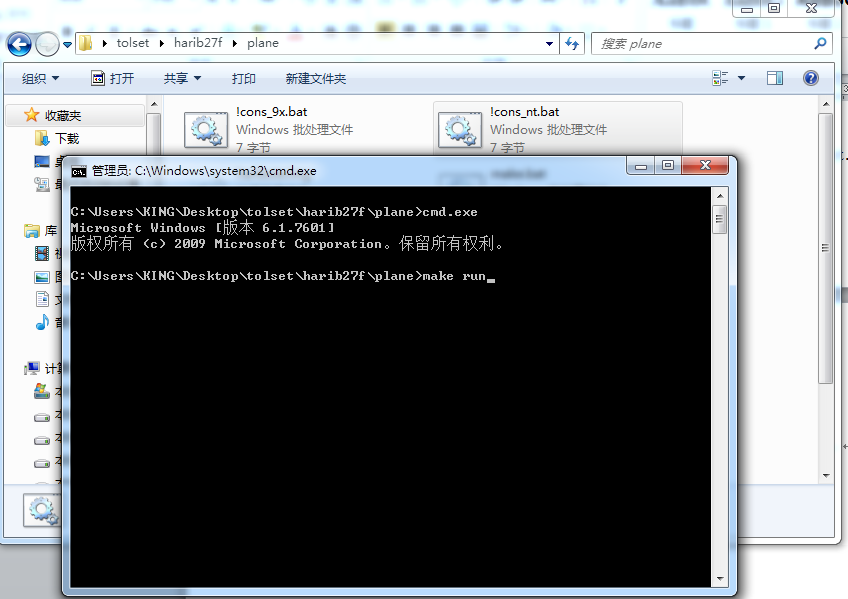
**七 程序运行示例**

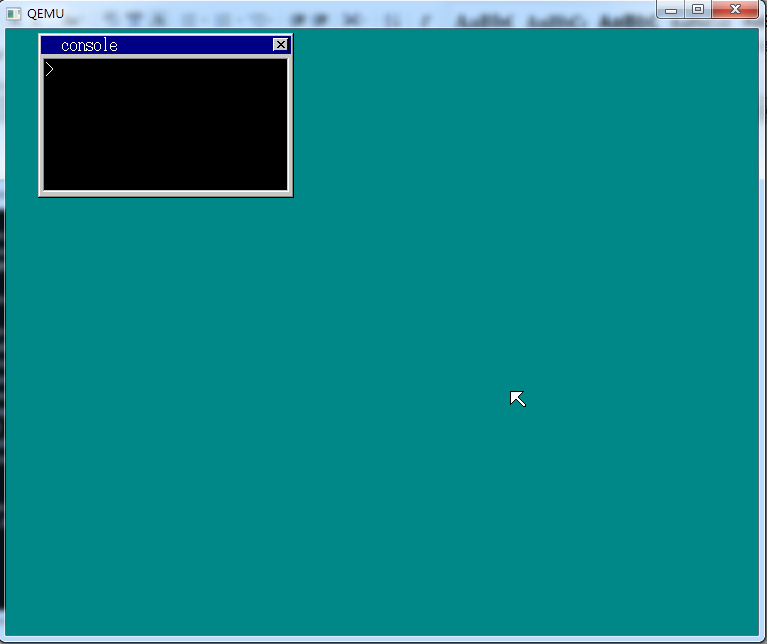
在运行程序前首先要运行操作系统。

运行说明

1.将项目文件夹拷贝到tolset文件夹下。

2.进入项目文件夹，找到plane文件夹。进入plane后运行“!cons\_nt.bat” 输入make run 启动操作系统





3.在操作系统中的console中输入plane，即可启动用户程序plane。



接触到陨石（@），提示GAME OVER



提示GAME OVER后，点击回车，会重新开始游戏。

**八 小结**

本次项目是3人小组，我们3人共同学习，独立开发，每个人都自己动手开发自己的应用，所以我们小组的项目有3个小游戏。我所开发是plane小游戏。

本次项目是操作系统课程的课程设计，加强了我们对操作系统的理解。但是不得不说，在这短短几天时间里掌握这些知识实在有些困难，尤其是参考源码一开始就是汇编语言，基本看不懂，有点被吓住。所幸最后动手写代码不需要很多的汇编知识，只需要了解一些即可。我们不仅学习了助教所给的于渊的教材《Orange’s一个操作系统的实现》，还找来了日本人川合秀实的教材《30天自制操作系统》。最终选择川合秀实的参考源码作为开发用户应用的基础，毕竟有图形界面更加用户友好，开发有界面的小游戏更优成就感。

经历的一番辛苦的学习，编码，测试，改BUG等等工作后，总算完成了预期的目标，实现了一个用户应用。当然，这个程序显然还有很多的改进空间，比如加入积分系统，发射子弹消灭“陨石”等，这些都考虑过，但是毕竟时间和精力有限，我相信如果有更多的时间，我的程序还可以更好。

2015-7-23