操作系统设计与功能说明文档

1.开发平台

Linux系统分支ubuntu,在ubuntu中使用bochs运行

2.开发人员及时间

1352897 蔡舒炘 01班

1352898 张昊 02班

1352904 李明阳 01班

完成周期：2015年7月20日－－2015年7月24日

3.开发目的

在参考源码的基础上实现一个用户级应用

4.功能说明

运行不需操作系统，有引导扇区，软盘引导。主要功能为实现一个经典的推箱子小游戏。要控制搬运工上下左右移动，来将箱子推到指定地点

5.实现细节

1.地图布局

地图上每个基本元素如：space、floor、box都用一个一个像素排列来显示

单元点坐标以单元为单位，常量 \*\_U

像素点坐标以像素为单位，常量 \*\_P

坐标点格式（x，y），先水平，后垂直

单元点的像素坐标为其右下角像素的像素坐标

水平 MAP\_HORZ\_U 个单元

垂直 MAP\_VERT\_U 个单元

单元点(xu,yu) in [0 .. MAP\_HORZ\_U)\*[0 .. MAP\_VERT\_U)

像素点(xp,yp) in [0 .. VIDEO\_HORZ\_P)\*[0 .. VIDEO\_VERT\_P)

二维图片实现3D效果，暂未使用，待扩充

绘制从左上角单元开始，至右下角单元结束，实现3D效果

地图原点为地图左上角单元(xu=0,yu=0)的像素坐标

2.函数功能

函数调用约定：

参数从右向左压栈，由调用函数平衡栈

被调用函数返回后与调用前相比，栈中元素及栈大小不改变

被调用方不维护 ax, bx, cx, dx，但保证维护其它寄存器

函数返回值 ax, bx, cx, dx

1）func void main()——主程序

2）func void shutdown()——关机

3）func void initOneLevel()——初始化 level 级别

4）func ax isUidU( uid, xu, yu )——判断单元点(xu,yu)是否为uid

ax!=0 是， ax==0 否

若此单元点不在地图中，返回 否

5）func ax isInMapU( xu, yu )——判断单元点(xu,yu)是否在地图中（所在点为SPACE算不在地图中）

ax!=0 在， ax==0 不在

6）func ax judgeWin()——判断是否胜利，ax==0 否，ax!=0 胜

7）func void eraseAll()——清除所有图像

8）func void drawAll()——直接重绘地图，无关背景，兼容3D效果，从左上角到右下角顺序绘制

9）func void drawUidU( uid, xu, yu )——绘制单元坐标为(xu,yu)的单元uid，uid == UID\_FLOOR

10）func ax, bx convU2P( xu, yu )，单元坐标转换为像素坐标，依据 orgx\_p, orgy\_p，返回 ax==xp, bx=yp，适应3D侧视图

11）func void drawUid( uid, xp, yp )，绘制像素点为(xp,yp)且类型为uid的单元点，uid == UID\_SPACE, UID\_FLOOR

12）func ax convUid2Uoff( uid )，uid == UID\_FLOOR, uoff = UOFF\_FLOOR, 不依赖 uid 和 uoff 的定义，无跳表，大量比较，处理了player的方向dir，BOX+DEST或BOX+FLOOR 按BOX处理。

13）func void drawUoff( uoff, xp, yp )——绘制像素点为(xp,yp)且地图单元位图偏移为uoff的单元，off == UOFF\_SPACE, UOFF\_FLOOR

14）func ax peekKey()——获取键盘输入，不阻塞，若无输入返回 VK\_NULL; 若有输入，取出，且不将其从缓冲区中清除

15）func ax getKey()——获取键盘输入，阻塞，若无输入，阻塞; 若有输入，取出，且将其从缓冲区中清除

16）func ax convKey2Vk( bios\_ax )——将BIOS键盘输入转换为虚拟键码，不依赖任何编码，大量比较

17）func void sleep( t )——等待 t 毫秒，等待中的输入被清除（清空输入缓冲区），移动动画使用

18）func void initVideo()——初始化显示方式

6.操作说明

提交的为源码（.asm文件），需用nasm box.asm -o box.bin 命令来转换成二进制文件，然后用dd if=box.bin of=a.img bs=512 count=15 conv=notrunc 命令来写入软盘。由于游戏本身较大，所以要将bochsrc文件中的megs改为64。

最后启动命令：bochs -f bochsrc。

游戏玩法

w、s、a、d分别为上下左右移动。r为重玩此关卡。q为结束游戏，跳转至end页面。

1. 实例演示

