

面向NEXYS4开发板的定制操作系统的设计与实现

指导老师：邵志远

课题目的与意义

- 课题来源：综合能力培养课程建设
- 设计目标：为NEXYS4开发板定制操作系统
- 国内外研究现况及发展趋势
 - 国外：MIT – JOS （开发环境Bochs、QEMU）
Harvard – OS161 （sys/161 machine simulator）
 - 国内：mips ucore：（QEMU）
 - 我们的选择：硬（NEXYS4 SoC）+软（ucore）
硬件和软件的融合

面临的挑战

- 设计难点与挑战：
 - 指令的替代与实现
 - 片上系统的TLB缺失异常处理
 - 片上系统定制外设与驱动实现
- 开发工具、调试环境：
 - 开发板环境，不具备GDB调试
 - 代码改动大、范围广，切换测试需大幅度调整

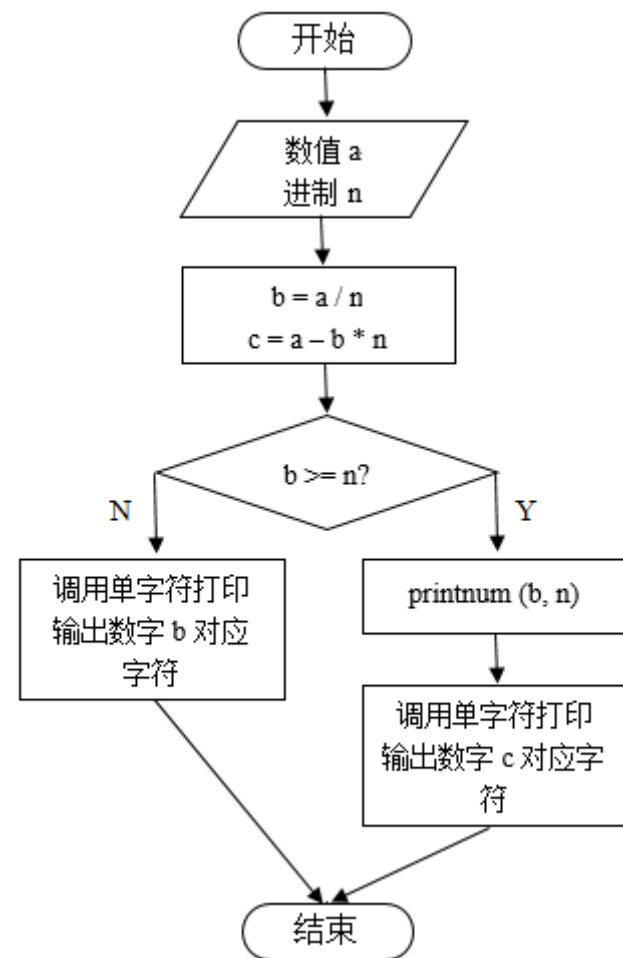
指令的替代与实现

- 编译后替代方式
- 乘法指令的替代与实现

方法	空间复杂度	时间复杂度
逐条加法	$O(n)$	$O(n)$
循环加法	$O(1)$	$O(n)$
移位+加法	$O(1)$	$O(1)$

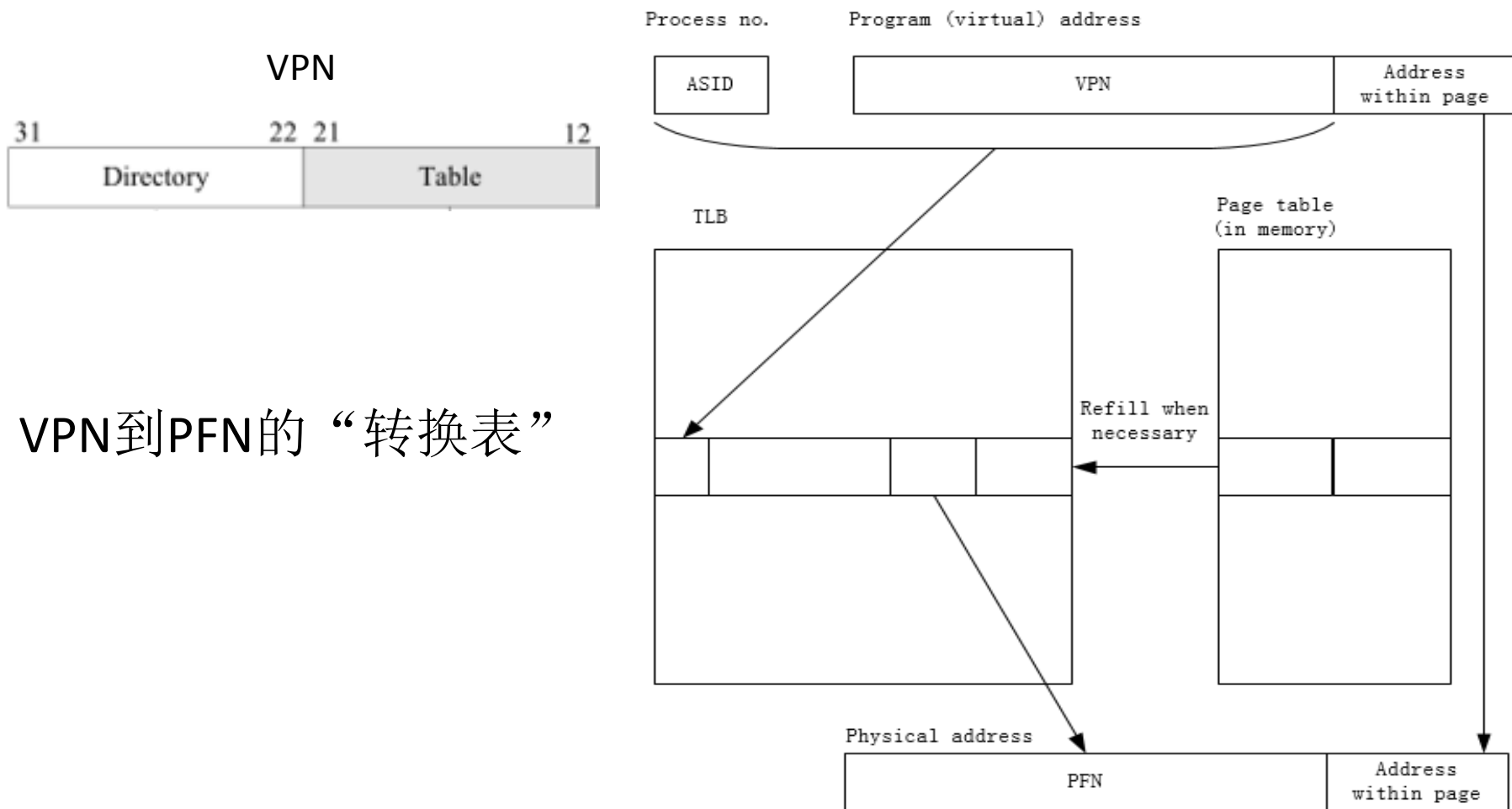
比较各方法的优化程度进行选择

- 除法指令的替代与实现
 - 涉及的内核函数kprintf、rand等
 - 其他的指令的替代与实现
- tlbwr等等



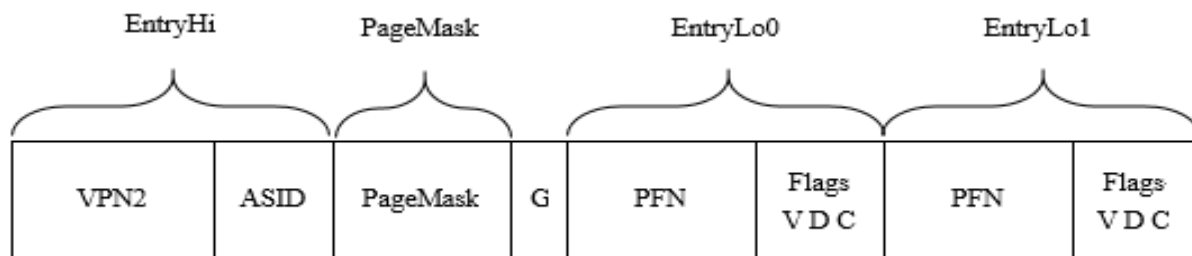
片上系统的TLB缺失异常处理

- 片上系统的页式地址转换机制



片上系统的TLB缺失异常处理

- 片上系统的TLB结构与相关CP0寄存器



一个虚拟页号对应两个物理页号

TLB中的一项

寄存器助记符	功能
EntryHi	具备VPN和ASID
EntryLo0 EntryLo1	VPN所映射到的物理页号，以及对应物理页的存取权限
PageMask	用来创建能映射超过 4KB 的页的入口
Index	决定相应指令要读写的TLB表项
Random	这个伪随机值（实际上是一个自由计数的计数器）用来让 tlbwr 写入新的 TLB 入口到一个随机选择的位置。

片上系统的TLB缺失异常处理

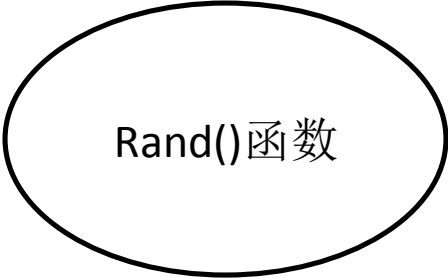
- 片上系统的TLB刷新机制
- 使用FIFO刷新方式代替随机刷新

算法3.1 TLB刷新算法的实现

输入：发生TLB缺失异常的地址：BadAddr

输出：NULL

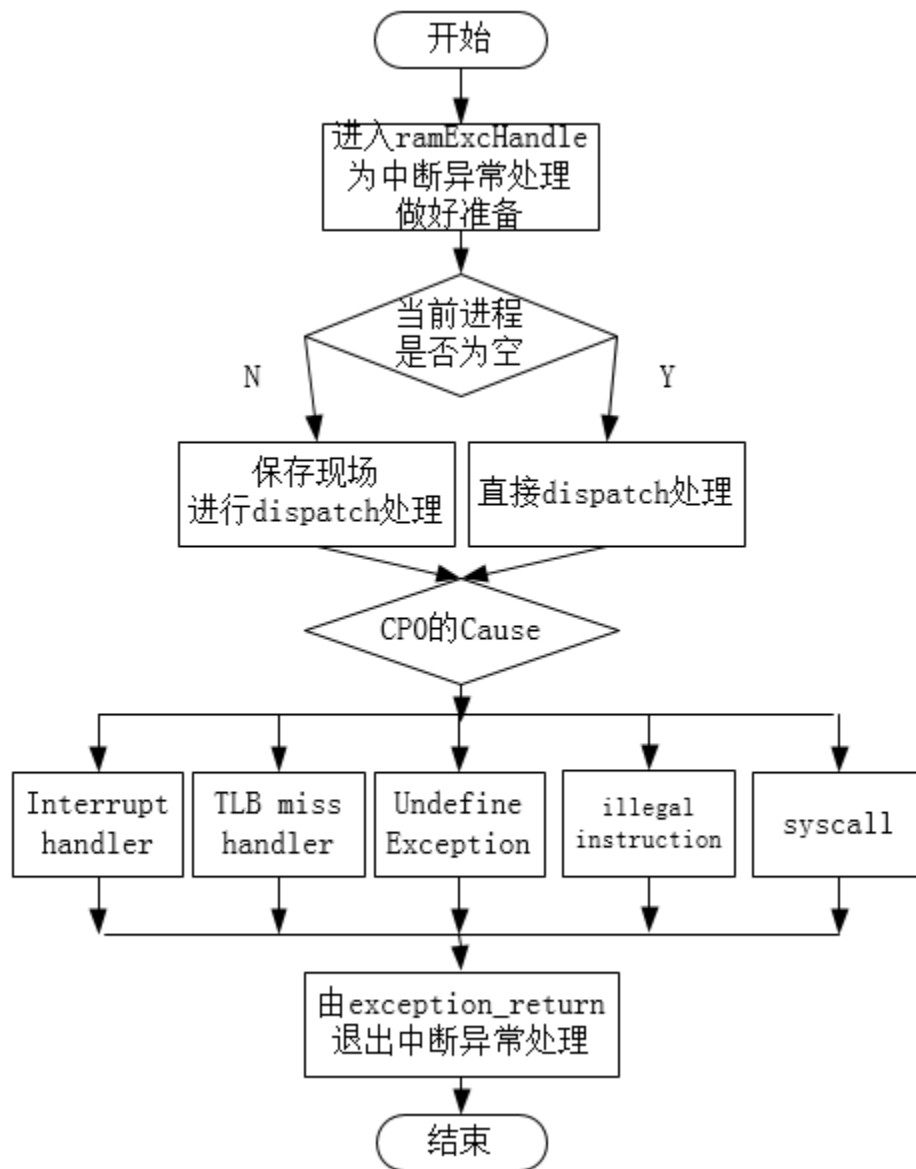
```
1:  #ifdef MACH_QEMU
2:      write_cp0_register(BadAddr) #将一些信息写入CP0的辅助寄存器
3:      __asm__ __volatile__ (“tlbwr”) #利用Random寄存器信息刷新TLB
4:  #elif defined MACH_FPGA
5:      static int index = 0;
6:      write_cp0_register(BadAddr) #将一些信息写入CP0的辅助寄存器
7:      write_c0_index(index++) #将index值写入CP0的Index寄存器
8:      __asm__ __volatile__ ( “tlbwi” ) #刷新由Index寄存器值指定的TLB项
9:  #endif
10:  return NULL
```



Rand()函数

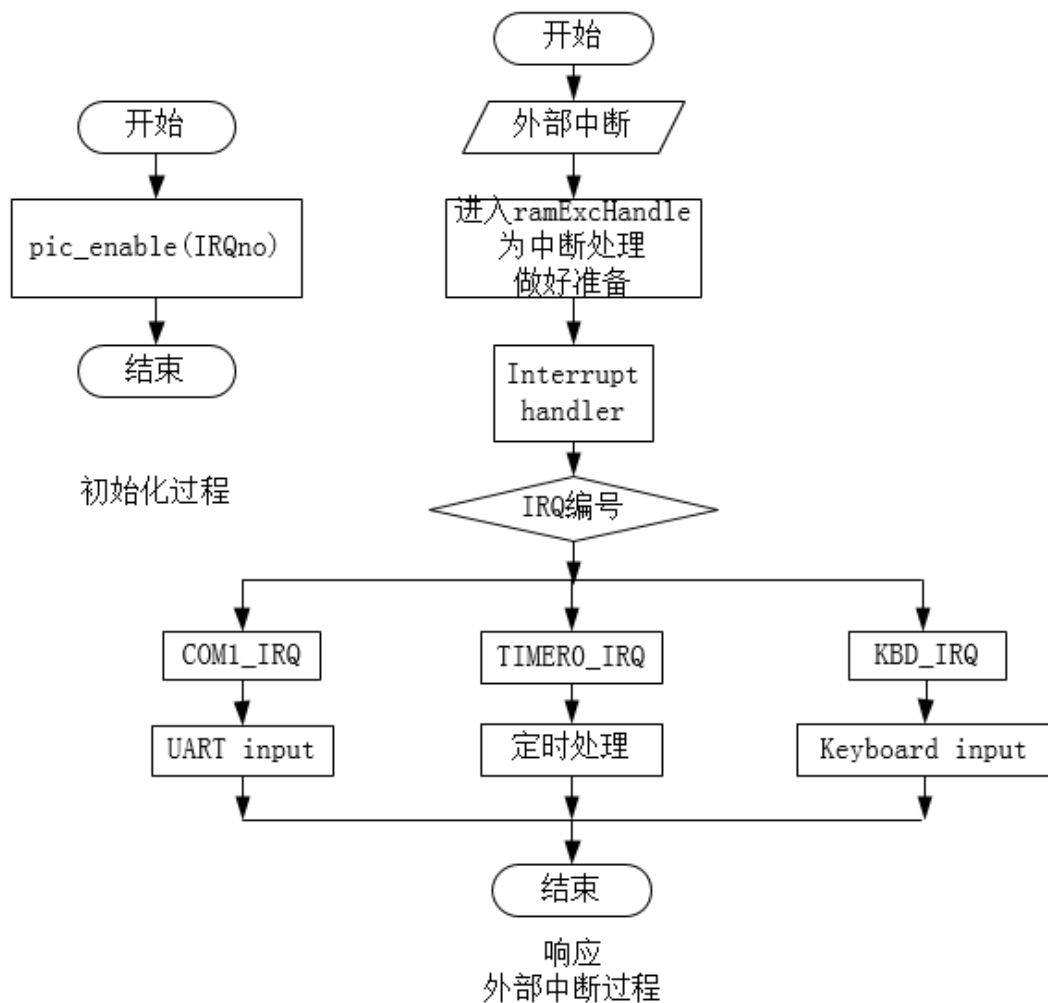
片上系统的缺页异常处理

- 中断和异常处理
- CP0相关寄存器
- Cause寄存器
- BadAddr寄存器
- EPC寄存器



片上系统定制外设与驱动实现

- 外设的分类
- 串口的输入设计
 1. 开启中断响应
 2. 如何响应中断
 3. 响应后如何进行处理



片上系统定制外设与驱动实现

- 串口的输出设计与实现

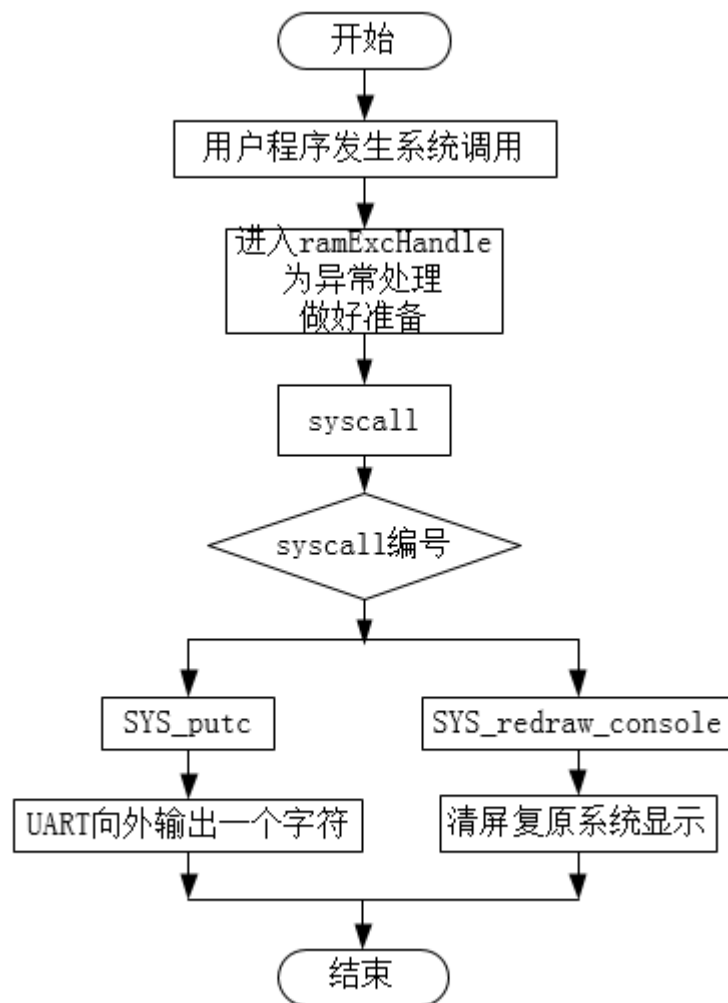
1. 自陷异常

2. 如何处理数据的输出

- Others : Keyboard、VGA

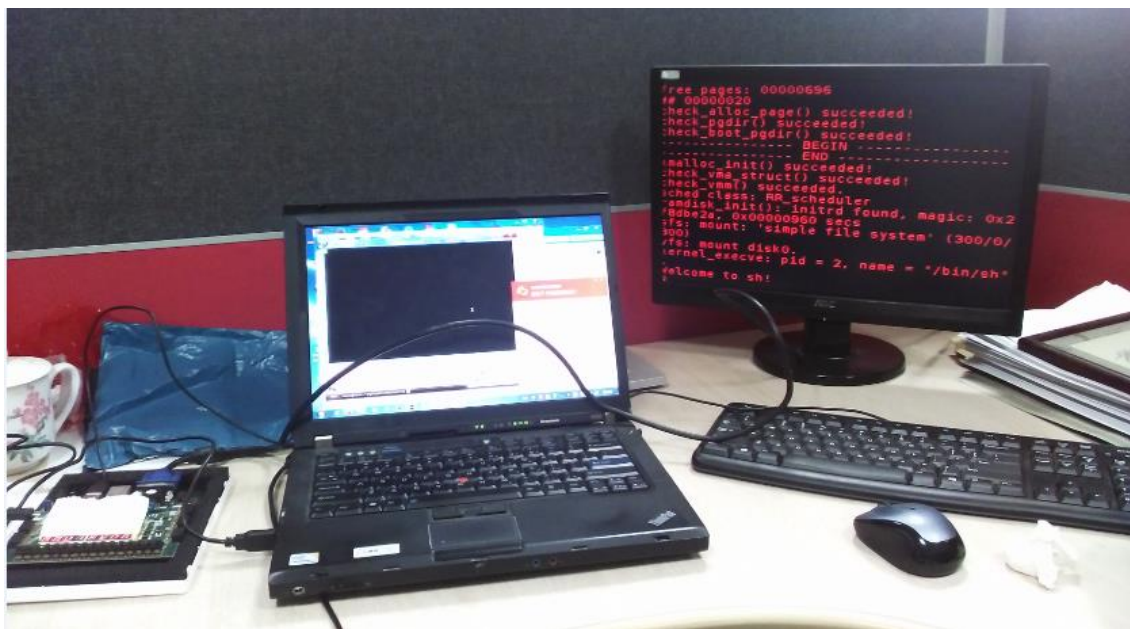
Keyboard -> 只读设备

VGA -> 只写设备



系统测试与结果

项目	信息说明
处理器	自制MIPS CPU @1.56MHz (1 CPU)
内存	8MB RAM
引导固件	4KB ROM
外设	UART16450 + PS/2键盘 + VGA文字模式



进入用户shell

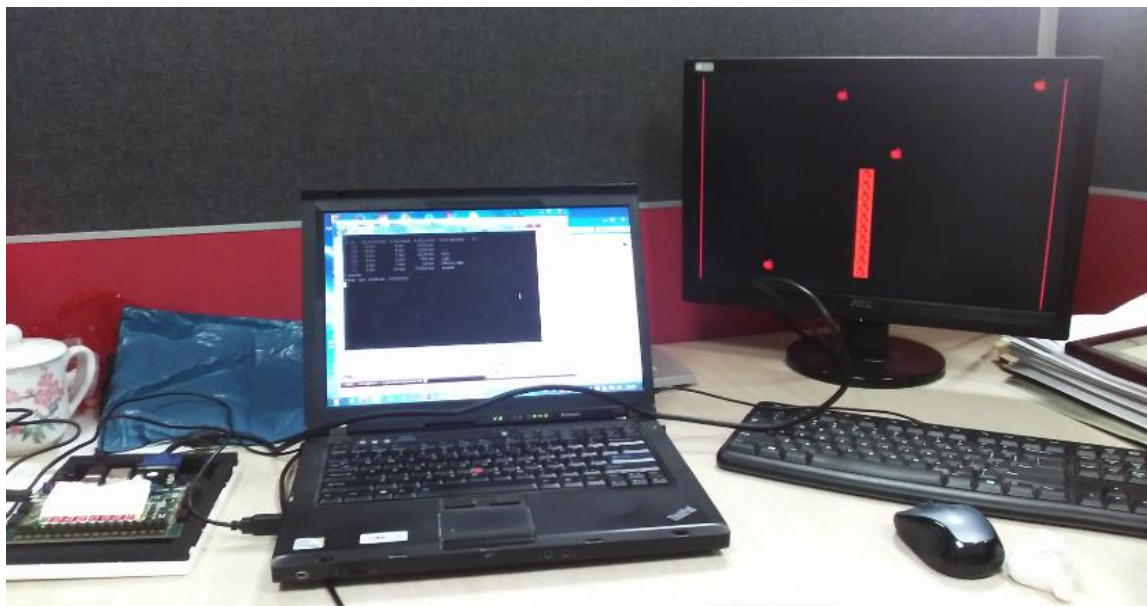
系统测试与结果

```
$ cd bin
$ ls
@ is [directory] 2(hlinks) 7(blocks) 2304(bytes) : @'.'
```

[d]	2(h)	7(b)	2304(s)	.
[d]	4(h)	3(b)	1280(s)	..
[-]	1(h)	30(b)	121384(s)	num
[-]	1(h)	31(b)	124540(s)	cp
[-]	1(h)	33(b)	132015(s)	ls
[-]	1(h)	29(b)	118026(s)	echo
[-]	1(h)	32(b)	127233(s)	pwd
[-]	1(h)	34(b)	136304(s)	sh
[-]	1(h)	29(b)	118381(s)	cat

一些基本
Unix命令

Retro Snaker
游戏



Thanks for Listening !

Q & A