

2021 年《计算机操作系统课程设计》

自测报告

题目名称	仿真实现 Linux/Unix 系统				
院系	人工智能学院	班级	计科 181	测试时间	2021/3/31
指导老师		助教姓名		电话 QQ	
姓名 学号		手机 QQ		申请成绩 (自行填写)	A
讨论组组长姓名		组长手机		讨论组成员姓名	
完成任务与功能说明	<p>1、计算机硬件仿真。</p> <p>实现部分。基础硬件：CPU，MMU，Clock（时钟）、Memory（时钟）、Disk（磁盘）。硬件细节仿真：IR（指令寄存器）、PC（程序计数器）、Register（通用寄存器）、Block（物理块）。个人拓展硬件仿真：SuperBlock（超级块）、GuideBlock（引导块）、RandomGenerator（随机数生成器）、Language（字符转换器）。</p> <p>功能。仿真实现计算机基础硬件。本程序中，磁盘的每一块由一个 txt 文件描述，并保存在 disk 文件夹下，每个文件夹每一行存储一个 4 位十六进制数，共 256 行，用来描述物理块中每字的数据，而对于其中存储的字符型数据（如文件名的存储），本系统采用额外编码，即使用 Language 转换。其他个人拓展部分将在之后环节中展开描述。</p> <p>2、文件管理</p> <p>实现及其功能。仿照了 Linux 文件系统进行设计。对磁盘进行分区，引导块，超级块，Inode 区，系统表区，交换区（作为内存中某进程的拓展），作业区，代码区，预留区（方便拓展系统功能）和文件区。超级块保存的信息为内存和磁盘 Inode 使用情况，内存块使用情况，文件区的空闲链表表头。文件的逻辑结构为顺序结构，文件的物理结构为索引分配，文件与 Inode 是一一对应的。内存空闲块，内存空闲 Inode，外存空闲 Inode 块为空闲表法，外存空闲块采用成组连接法。系统中有 i_Inode 表（存放磁盘中所有 inode 的索引信息），系统打开文件表，进程打开文件表，完成了打开，读写，关闭文件的系统调用。</p> <p>3、内存管理</p> <p>实现及其功能。MMU 可实现地址变换，缺页中断处理，页表置换策略为 LRU，同时采用虚拟页式存储。</p> <p>4、作业管理与进程调度</p> <p>实现了读入预留作业，新建随机、半随机作业的功能。作业调度采用静态优先级优先+先来后到的调度策略。进程调度采用静态级+时间片轮转，采用五态转换模型，并在设计上有一定的死锁预防策略。根据指令设计，又设计了 5 中阻塞类型，文件读写系统调用阻塞+占用缓冲区资源（资源数远大于阻塞时间），键盘输入与屏幕输出阻塞+占用键盘/屏幕资源（可根据指令操作数设置阻塞时间），打印机阻塞+占用打印机资源（得不到资源的会进入打印机等待阻塞），摄像头阻</p>				

塞+占用摄像头资源。

同时设计了指令，其格式为类型+操作数。具体如下。

指令类型	操作数	功能
0	0~5	Value: 声明一个变量。产生一个不产生缺页中断的逻辑地址，并进行寻址。操作数表示变量类型，
1	0~3	Calculate: 计算。产生一个覆盖所有逻辑地址的随机数，并进行寻址。操作数表示计算方式。
2	0~9	Read: 读文件。打开并读文件（inode 号为操作数），两秒后关闭文件。
3	0~9	Write: 写文件。打开并写文件（inode 号为操作数），两秒后关闭文件。
4	5~15	Input: 键盘输入。输入时间由操作数指定。
5	5~15	Output: 屏幕输出。输出时间由操作数指定。
6	0	Printer: 打印机服务。操作数无意义。
7	0	Camera: 摄像头服务。操作数无意义。

5、设备管理

系统有一张设备表，设备号隐含为表项号，记录设备名，设备驱动程序地址以及各个设备的有效工作时间。采用中断方式，每个设备一个线程，并由时钟中断唤醒一次工作周期。

6、GUI 可视化设计

设计了 GUI 界面。有开机界面，有系统信息显示主界面，同时也可以由主界面打开进入内存管理、磁盘管理、设备管理、进程管理、作业管理、文件管理、操作系统进程界面。同时也可以实现重装系统和暂停/恢复系统时钟的功能。

1、环境配置

安装一个 JavaSE-15 版本的 JDK。并配置 JAVA_HOME 与 PATH 到用户变量中。

2、目录配置

保证 jar 文件目录为以下格式。

-执行文件夹

-disk

-input

-output

-19218101_马树凡_OS.jar

其中 disk, input, output 文件夹已随附到文件中。

此电脑 > 本地磁盘 (D:) > 学习 > 桌面实验习题文件夹 > 操作系统课设 > 最终 > 执行

名称	修改日期	类型	大小
disk	2021/4/1 8:46	文件夹	
input	2021/4/1 8:46	文件夹	
output	2021/4/1 8:44	文件夹	
19218101_马树凡_OS.jar	2021/4/1 8:43	JAR 文件	246 KB

3、运行 jar 文件

Win10 系统下进入 CMD 中，定位到执行文件夹。

输入 java -jar 19218101_马树凡_OS.jar 即可运行。



4、GUI 功能说明

首先进入首界面，等待开机完成后输入密码 123456，点击确定。

进入主界面后，点击“新建作业”可以进行创建作业，其中“创建”可指定作业类型，优先级，代码数量，“随机创建”则全部随机。

点击“作业管理”、“内存管理”、“磁盘管理”、“进程管理”、“文件管理”、“设备管理”可以查看各种系统信息。

点击“操作系统进程”可以查看操作系统进程的相关信息。

点击“关机”则关闭。

注意每种打开的界面，必须点击“关闭”按钮才能关闭。

（按照拟完成功能模块和函数逐项测试，写清楚测试用例使用的输入数据、输入方式、界面和输出结果界面与内容，并对每个测试用例输出结果的正确性加以分析并说明。测试用例需要体现本题目的核心技术点、测试数据需要多组并有一定代表性。已给的项目必须测试填写，在此基础上扩展）

1、作业描述文件

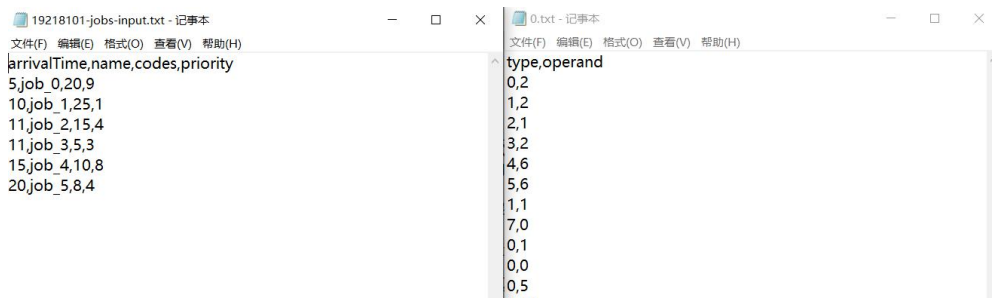
在 input 文件夹下，存储着初始作业集 19218101-jobs-input.txt 文件，其中存储格式如下（每一列用英文逗号相隔）：

arrivalTime	Name	Codes	Priority
5	Job_0	20	9
.....			

每一项作业号隐藏为其顺序号（从 0 开始），同时每一项作业对应的代码内容存储在“作业号.txt”文件中，存储格式如下（地址隐含，为从 0 开始的顺序地址）：

Type	Operand
0	2
.....	

示例描述文件如下：



2、并发作业请求（初始作业集）

测试用例的初始作业集共 6 条作业，用于如下测试，具体内容如下：

作业集文件

ArrivalTime	Name	Codes	Priority
5	Job_0	14	9
5	Job_1	10	5
7	Job_2	10	5
8	Job_3	15	5
9	Job_4	10	5
15	Job_5	5	6
20	Job_6	5	1

作业 0 代码文件

Type	Operand	Type	Operrand
0	1	1	2
0	2	1	3
0	3	0	5
0	4	0	2
0	5	2	4

功 能 及 性 能 测 试	1	0	3	6
	1	1	4	12
	作业 1 代码文件			
	Type	Operand	Type	Operand
	2	3	6	0
	3	5	6	0
	2	4	1	0
	6	0	2	7
	6	0	5	8
	作业 2 代码文件			
	Type	Operand	Type	Operand
	2	3	6	0
	3	5	6	0
	2	4	6	0
	0	1	1	0
	6	0	2	7
	作业 3 代码文件			
	Type	Operand	Type	Operand
	2	3	6	0
	3	5	1	0
	2	4	2	7
	0	1	5	8
	1	3	1	2
	6	0	0	2
	6	0	0	3
	6	0		
	作业 4 代码文件			
	Type	Operand	Type	Operand
	2	3	0	2
	3	5	6	0
	2	4	6	0
	0	1	6	0
	1	3	0	2
	作业 5 代码文件			
	Type	Operand	Type	Operand
	2	3	5	15
	3	4	6	0
	4	15		
	作业 6 代码文件			
	Type	Operand	Type	Operand
	6	0	6	0
	7	0	0	1
	7	0		

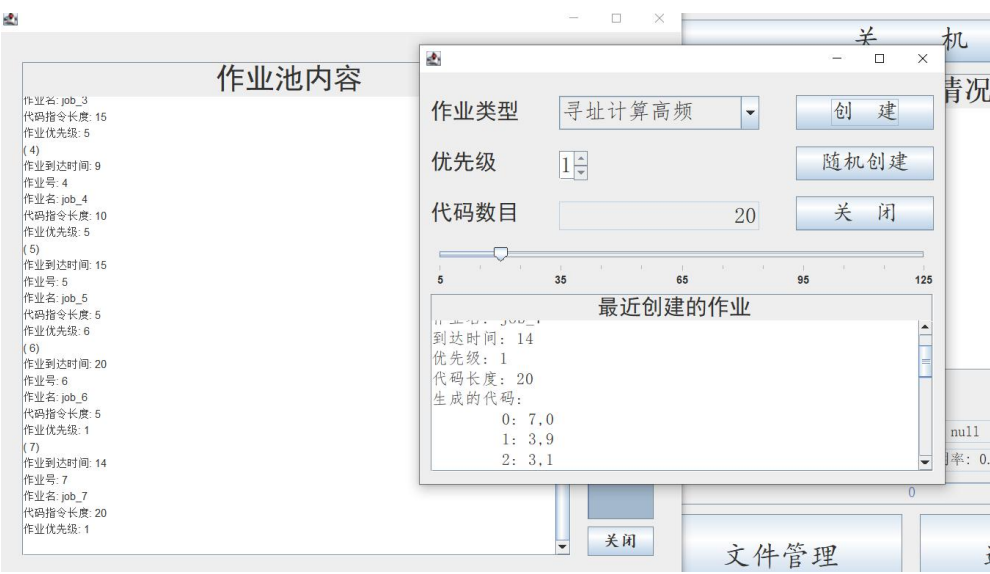
加载到系统的信息显示



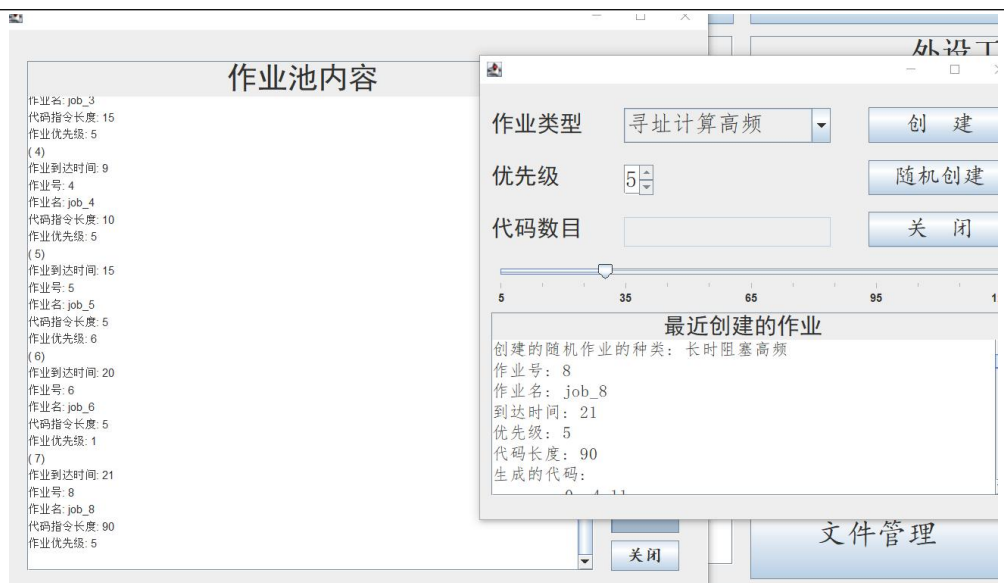
初始作业集设计原则，0 号作业为高频寻址作业类型，1 号与 0 号作业检验静态优先级的作业调度算法，1,2,3,4 号作业有一段连续的打印机请求服务，用来检查当打印机资源空时，打印机等待阻塞的情况，5,6 号作业为长时间阻塞的作业类型。

3、并发作业请求（随机作业产生）

半随机模式，设置“寻址计算高频”作业类型，优先级为 1，代码长度为 20。作业池中出现新个作业，创建作业界面中出现创建的作业的全部信息。



全随机模式，直接点击随即创建，作业池中出现新的作业，创建作业界面中出现创建的作业的全部信息。



4、MMU 地址变化

以系统执行作业 0 的第一条指令为例。

```

5: 0 号PCB新建态 -> 就绪态
5: 0 号PCB就绪态 -> 运行态
5: 0 号进程执行一条指令
    ●PC: 1 Type: value Operand: short
    ●逻辑页号: 0, 内存块号: 28, 外存块号: 128, 偏移地址: 230
    ●PCBIndex: 0, 访问地址(230 -> 7398)
6: 开始运行 1 号作业
6: 1 号PCB被创建为新建态
    ●进程名(job_1), 代码数(10), 优先级(5)
6: 1 号PCB新建态 -> 就绪态
6: 0 号进程执行一条指令
  
```

预分配时将 0,1 号页面调入内存，而 value 类型的指令是产生不引发缺页中断的随机地址，这里可以看出访问地址为 230（0 页），而 0 号进程为第一个被调入内存中的作业，其分配的内存块空间为 16（代码块），28, 29（数据块）块，所以将其地址变换为：内存块号 * 256 + 偏移地址 = 7398。

5、MMU 缺页中断

以系统执行作业一的某一条指令为例。

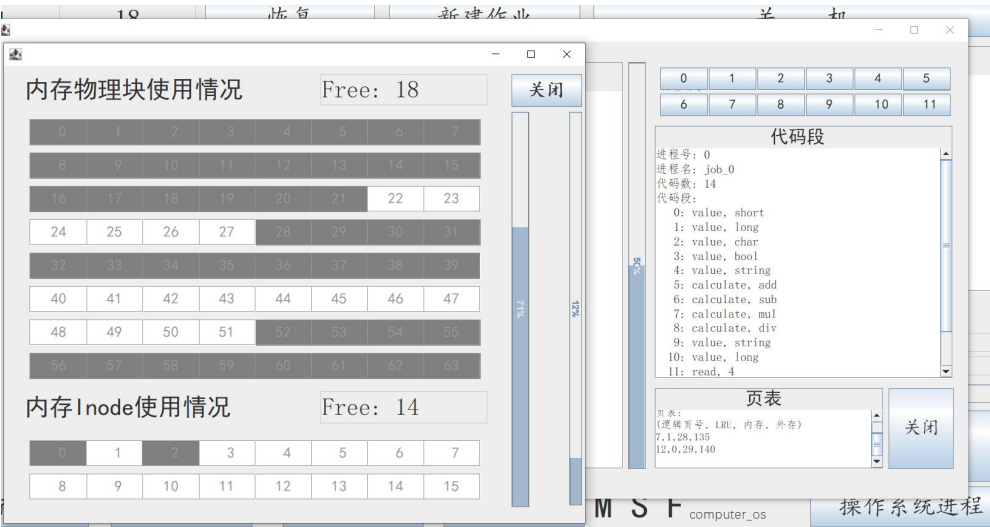
```

9: 0 号进程执行一条指令
    ●PC: 5 Type: value Operand: string
    ●逻辑页号: 1, 内存块号: 29, 外存块号: 129, 偏移地址: 28
    ●PCBIndex: 0, 访问地址(284 -> 7452)
10: 0 号进程执行一条指令
    ●PC: 6 Type: calculate Operand: add
    ●(0, 28, 128) -> (27, 28, 155)
    ●逻辑页号: 27, 内存块号: 28, 外存块号: 155, 偏移地址: 184
    ●PCBIndex: 0, 访问地址(7096 -> 7352)
11: 0 号进程执行一条指令
    ●PC: 7 Type: calculate Operand: sub
  
```

10 时钟时间时执行 calculate 指令，该指令为产生一个进程逻辑地址空间（0~8191）的随机数以供寻址，大概率引发缺页中断，可以观察到，上一条指令访问了 1 号页面，所以在缺页中断时选择替换 0 号页面（LRU 算法），而要访问的逻辑地址为 7096，其逻辑页号为 $7096/256=27$ ，所以 0 号替换为 27 号页面，之后的内存地址变换与 MMU 地址变化所述一致。

6、内存空间分配

某时刻系统进程数为 6 的分配。



内存物理块 16~21 被分配给 6 个进程的代码块（每个进程一块），28~39 被分配给 6 个进程的数据块（每个进程 2 块），同时该时刻还有一次文件读写指令，加上操作系统常置打开主目录，共分配了两个 Inode。

7、页表生成

每个进程被创建时会分配 0,1 逻辑页号。



之后随着寻址访问更新页表，比如寻址访问频繁的 0 号作业。



8、作业调度

根据测试用例，可以得到 0 号作业最先调度，其优先级在已到达的时间里最高。

计算机内部

0: CPU空转(无运行态)
1: CPU空转(无运行态)
2: CPU空转(无运行态)
3: CPU空转(无运行态)
4: CPU空转(无运行态)
5: 开始运行 0 号作业
5: 0 号PCB被创建为新建立态
 ● 进程名(job_0)，代码数(14)，优先级(9)
5: CPU空转(无运行态)

(0)
作业到达时间: 5
作业号: 1
作业名: job_1
代码指令长度: 10
作业优先级: 5

(1)
作业到达时间: 7
作业号: 2
作业名: job_2
代码指令长度: 10
作业优先级: 5

9、进程调度

全部进程调度信息会在计算机内部关键信息中显示。

计算机内部关键信息

11: 0 号进程执行一条指令
 ● PC: 7 Type: calculate Operand: sub
 ● (1, 29, 129) -> (12, 29, 140)
 ● 逻辑页号: 12, 内存块号: 29, 外存块号: 140, 偏移地址: 101
 ● PCBIndex: 0, 访问地址(3173 -> 7525)
12: 0 号进程执行一条指令
 ● PC: 8 Type: calculate Operand: mul
 ● (10, 28, 138) -> (7, 28, 135)
 ● 逻辑页号: 7, 内存块号: 28, 外存块号: 135, 偏移地址: 119
 ● PCBIndex: 0, 访问地址(1911 -> 7287)
13: 0 号进程执行一条指令
 ● PC: 9 Type: calculate Operand: div
 ● 逻辑页号: 12, 内存块号: 29, 外存块号: 140, 偏移地址: 164
 ● PCBIndex: 0, 访问地址(3236 -> 7588)
14: 0 号进程执行一条指令
 ● PC: 10 Type: value Operand: string
 ● 逻辑页号: 7, 内存块号: 28, 外存块号: 135, 偏移地址: 200
 ● PCBIndex: 0, 访问地址(1992 -> 7368)
15: 开始运行 5 号作业
15: 5 号PCB被创建为新建立态
 ● 进程名(job_5), 代码数(5), 优先级(6)
15: 5 号PCB新建态 -> 就绪态
15: 0 号进程执行一条指令
 ● PC: 11 Type: value Operand: long
 ● 逻辑页号: 7, 内存块号: 28, 外存块号: 135, 偏移地址: 254
 ● PCBIndex: 0, 访问地址(2046 -> 7422)
16: 0 号进程执行一条指令

外设工作情况

0: Read/Write磁盘工作空转
0: 键盘工作空转
0: 屏幕工作空转
0: 打印机工作空转
0: 摄像头工作空转
1: Read/Write磁盘工作空转
1: 键盘工作空转
1: 屏幕工作空转
1: 打印机工作空转
1: 摄像头工作空转
2: Read/Write磁盘工作空转
2: 键盘工作空转
2: 屏幕工作空转
2: 打印机工作空转
2: 摄像头工作空转

CPU
PC: 1 IR: read, 3
状态: 内核态 利用率: 61.11%

61

文件管理 进程管理

10、进程基础信息查看

可以查看五态的所有进程信息，阻塞态会额外包括保护现场信息。在界面右边可以点击进程池的索引号,来查看该索引号对应的 PCB 中存储的进程基础信息，包括其代码和页表。中间进度条会显示进程池的使用情况。

新建态

就绪态

运行态

阻塞态

已完成

0: 开始时间: 5
运行时间: 11
进程号: 0
进程名: job_0
优先级: 9
代码长度: 14
页表:
(逻辑页号, LRU, 内存, 外存)
7,1,28,135
12,0,29,140
1:
开始时间: 6

0: 开始时间: 15
运行时间: 0
进程号: 5
进程名: job_5
优先级: 6
代码长度: 5
页表:
(逻辑页号, LRU, 内存, 外存)
0,1,38,288
1,0,39,289
现场:
CPU: 内核态

0 1 2 3 4 5
6 7 8 9 10 11

代码段

进程号: 5
进程名: job_5
代码数: 5
代码段:
0: read, 3
1: write, 4
2: input, 15
3: output, 15
4: printer, 0

页表

页表:
(逻辑页号, LRU, 内存, 外存)
0,1,38,288
1,0,39,289

关闭

11、程序运行调度输出 ProcessResults.txt 文件内容

全部运行信息保存在此文件中。

0: CPU空转(无运行态)
0: Read/Write磁盘工作空转
0: 键盘工作空转
0: 屏幕工作空转
0: 打印机工作空转
0: 摄像头工作空转
1: CPU空转(无运行态)
1: Read/Write磁盘工作空转
1: 键盘工作空转
1: 屏幕工作空转
1: 打印机工作空转
1: 摄像头工作空转
2: CPU空转(无运行态)
2: Read/Write磁盘工作空转
2: 键盘工作空转
2: 屏幕工作空转
2: 打印机工作空转
2: 摄像头工作空转

12、界面程序运行信息显示

进程调度信息和设备使用情况，此外还会显示 CPU 利用率，CPU 当前内容。

时钟86

恢复

新建作业

关机

计算机内部关键信息

0: CPU空转(无运行态)
1: CPU空转(无运行态)
2: CPU空转(无运行态)
3: CPU空转(无运行态)
4: CPU空转(无运行态)
5: 开始运行 0 号作业
5: 0 号PCB被创建为新建态
 ●进程名(Job 0), 代码数(14), 优先级(9)
5: 0 号PCB新建态 -> 就绪态
5: 0 号PCB就绪态 -> 运行态
5: 0 号进程执行一条指令
 ●PC: 1 Type: value Operand: short
 ●逻辑页号: 0, 内存块号: 28, 外存块号: 128, 偏移地址: 230
 ●PCIndex: 0, 访问地址(230 -> 7398)
6: 开始运行 1 号作业
6: 1 号PCB被创建为新建态
 ●进程名(Job 1), 代码数(10), 优先级(5)
6: 1 号PCB新建态 -> 就绪态
6: 0 号进程执行一条指令
 ●PC: 2 Type: value Operand: long
 ●逻辑页号: 0, 内存块号: 28, 外存块号: 128, 偏移地址: 206
 ●PCIndex: 0, 访问地址(206 -> 7374)
7: 开始运行 2 号作业
7: 2 号PCB被创建为新建态
 ●进程名(Job 2), 代码数(10), 优先级(5)
7: 2 号PCB新建态 -> 就绪态
7: 0 号进程执行一条指令

外设工作情况

0: Read/Write磁盘工作空转
0: 键盘工作空转
0: 屏幕工作空转
0: 打印机工作空转
0: 摄像头工作空转
1: Read/Write磁盘工作空转
1: 键盘工作空转
1: 屏幕工作空转
1: 打印机工作空转
1: 摄像头工作空转
2: Read/Write磁盘工作空转
2: 键盘工作空转
2: 屏幕工作空转
2: 打印机工作空转
2: 摄像头工作空转

CPU

PC: 10IR: value, long

状态: 内核态利用率: 27.91%

27

文件管理

进程管理

13、文件 Inode 信息的查看

文件管理中可以点击磁盘中的 inode 号来查看其具体内容。

《计算机操作系统课设设计》测试报告

南京农业大学人工智能学院计算机科学与技术系

第 10 页

Inodes	文件名: 外部设备驱动程序
0	Inode号: 2
1	Inode位置: 2
2	文件类型: DIR
3	读权限: true
4	写权限: true
5	用权限: true
6	创建时间: 2021/1/1-12:43
7	修改时间: 2021/3/1-17:38
8	文件大小: 512B
9	直接地址:
10	2054
11	一级间接地址:
12	null

选择
关闭界面

Inodes	文件名: OStest_0	进程数:
0	Inode号: 11	
1	Inode位置: 11	
2	文件类型: TXT	
3	读权限: true	
4	写权限: true	
5	用权限: true	
6	创建时间: 2021/3/2-14:30	
7	修改时间: 2021/3/29-10:11	
8	文件大小: 512B	
9	直接地址:	
10	2067	
11	一级间接地址:	
12	null	

选择
关闭界面

14、系统打开文件表

55 时钟时，执行 2 号进程的 read 指令。

●PC: 9 Type: read Operand: 7

53: 1 号PCB运行态 -> 阻塞态(文件读写队列)

53: 2 号进程由(打印机队列)阻塞态 -> 就绪态

54: 2 号PCB就绪态 -> 运行态

54: 2 号进程执行一条指令

●PC: 9 Type: calculate Operand: add

●(1, 33, 193) -> (21, 33, 213)

●逻辑页号: 21, 内存块号: 33, 外存块号: 213, 偏移地址: 153

●PCIndex: 2, 访问地址(5529 -> 8601)

54: 1 号进程由(文件读写队列)阻塞态 -> 就绪态

55: 2 号进程执行一条指令

●PC: 10 Type: read Operand: 7

55: 2 号PCB运行态 -> 阻塞态(文件读写队列)

2: 打印机工作空闲

2: 摄像头工作空闲

CPU

PC: 10 IR: read, 7

状态: 内核态 利用率: 28.57%

28

文件管理
进程管理

“文件管理”中查看系统文件打开表以及相应的 Inode 调入内存中的分配情况。

系统文件打开表

系统打开文件总数: 2

(0)	Inode号: 0	Inode在内存Inode分区的位置: 0(0, 0)	内存引用数: 1	是否修改: false
(1)	Inode号: 7	Inode在内存Inode分区的位置: 1(0, 1)	内存引用数: 1	是否修改: false

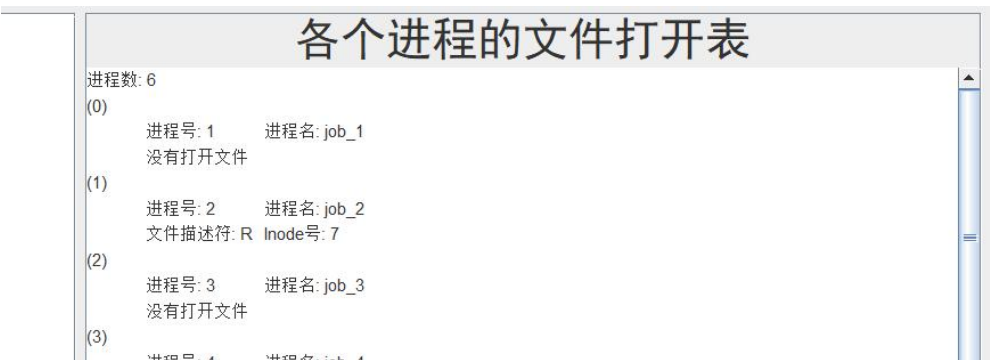
内存Inode使用情况

Free: 14

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15

15、进程打开文件表

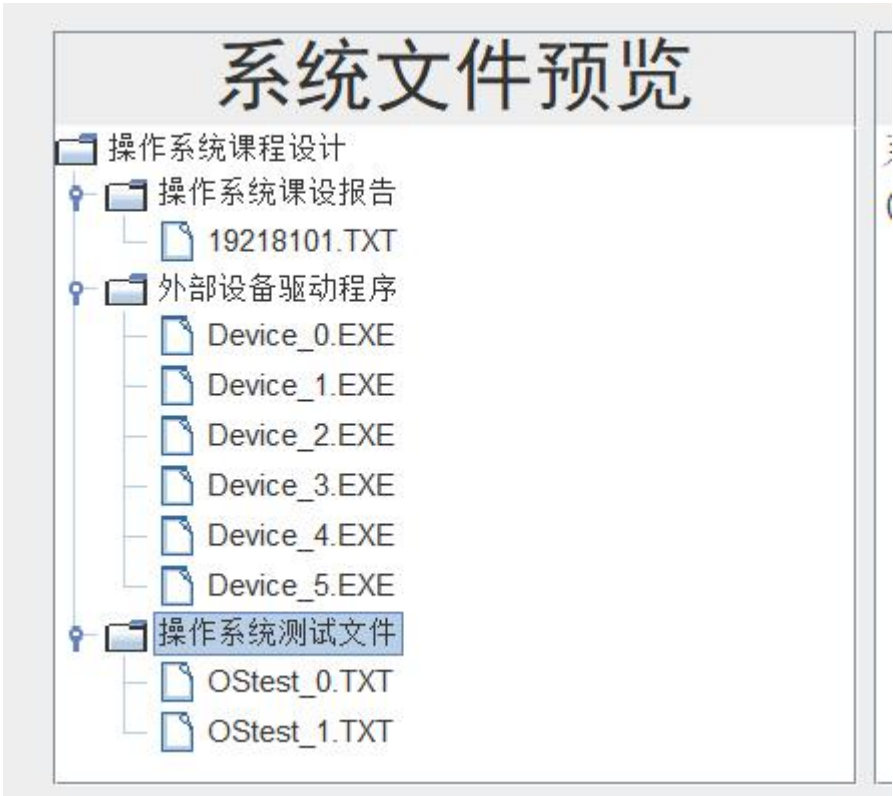
还是在 55 点钟刻，2 号进程打开文件表



16、文件系统的目录结构

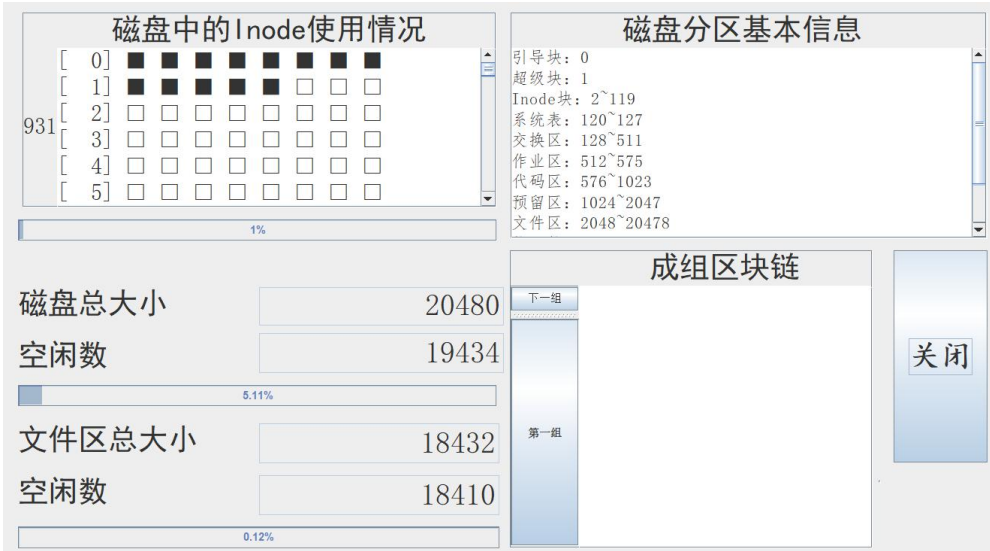
“文件管理”中可以查看文件系统的目录结构。





17、磁盘存储信息

显示磁盘 Inode 的位示图，以及使用 and 分区情况。



18、成组链接法

第一组空闲区块链。

20480

19434

5.11%

18432

18410

0.12%

下一组

第一组

成组区块链

Num: 43

Free: 2111, 2110, 2109, 2108, 2107, 2106, 2105, 2104, 2103, 2102

关闭

搜索到下一组。

20480

19434

5.11%

18432

18410

0.12%

下一组

第一组

成组区块链

Num: 64

Free: 2175, 2174, 2173, 2172, 2171, 2170, 2169, 2168, 2167, 2166

关闭

19、设备管理

在“设备管理”中，会显示所有外设的有效工作时间比，还有各种资源的剩余数目。同时还有外部设备表。

文件调用

剩余资源: 12, 总资源数: 12

打印机

剩余资源: 1, 总资源数: 3

(0) PCB号: 1, 优先级: 5, 到达阻塞时间: 0, 经过阻塞时间: 4, 应当阻塞时间: 6, 对应打印机: A(1), PCB号: 2, 优先级: 5, 到达阻塞时间: 0, 经过阻塞时间: 2, 应当阻塞时间: 6

键盘输入

剩余资源: 0, 总资源数: 1

(0) PCB号: 5, 优先级: 6, 到达阻塞时间: 0, 经过阻塞时间: 15, 应当阻塞时间: 15

摄像机

剩余资源: 1, 总资源数: 1

屏幕输出

剩余资源: 1, 总资源数: 1

等待打印机队列

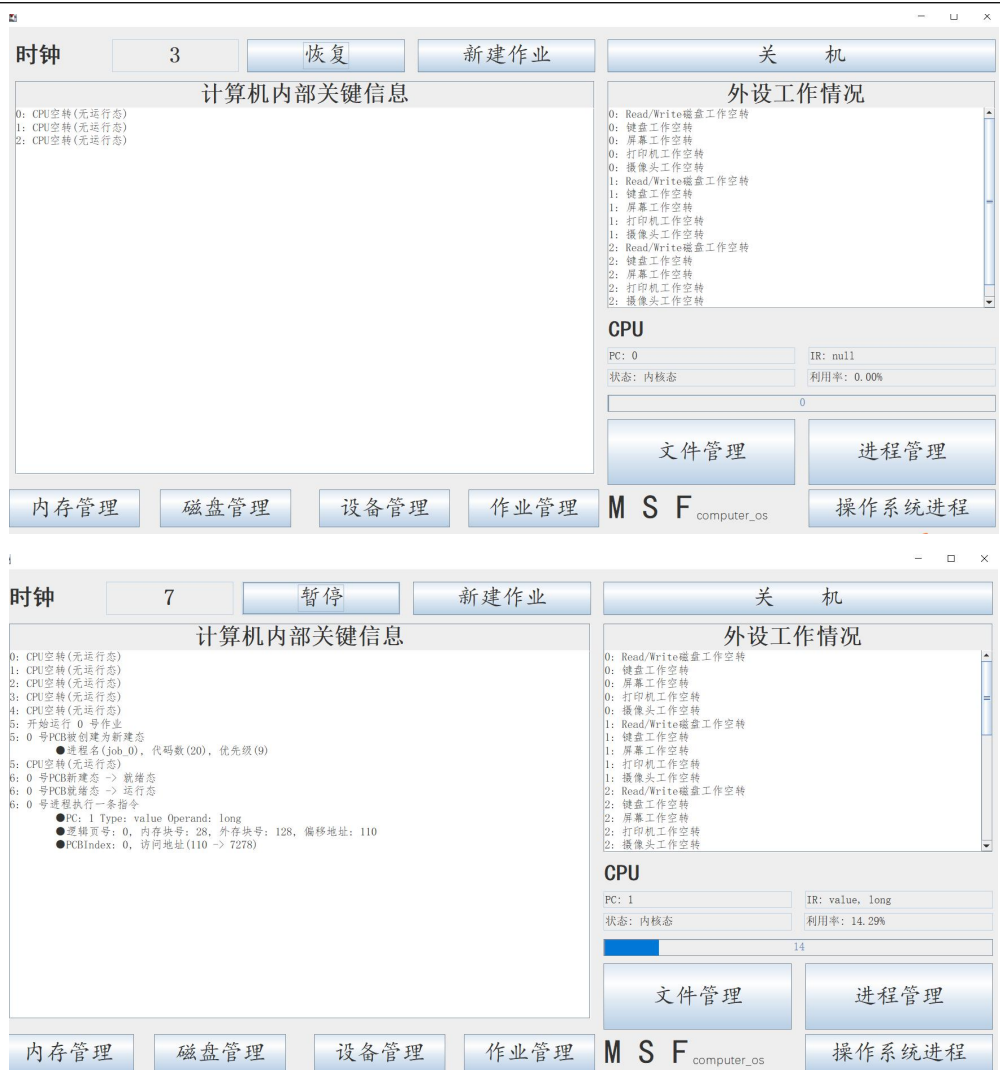
A打印机: 26%, B打印机: 21%, C打印机: 0%

关闭

设备号 设备名 入口
0 Logi_K380 2060H
1 Dell_LED 2061H
2 PrinterA 2062H
3 PrinterB 2063H
4 PrinterC 2064H
5 Lenovo 2065H

20、时钟暂停

暂停和恢复按钮



21、GUI 开机功能

开机未完成

开机信息

本地txt开始装载到模拟磁盘
装载0号柱面完成
装载1号柱面完成
装载2号柱面完成

马树凡

操作系统课程设计
2021.3.31

用户名

Student

密 码

默认用户名为任意字符
默认密码为123456

确 定

取 消

重装系统

开机完成（按钮变化）

开机信息

本地txt开始装载到模拟磁盘
装载0号柱面完成
装载1号柱面完成
装载2号柱面完成
装载3号柱面完成
装载4号柱面完成
装载5号柱面完成
装载6号柱面完成
装载7号柱面完成
装载8号柱面完成
装载9号柱面完成
本地txt装载到模拟磁盘完毕
作业已经加载成功
超级块已经装入内存中
i_InodeTable创建成功
DeviceTable初始化成功
操作系统已装入内存中
操作系统进程已被创建
OpenFileTable创建成功
主目录被加载到内存中

马树凡

操作系统课程设计
2021.3.31

用户名

Student

密 码

默认用户名为任意字符
默认密码为123456

确 定

取 消

重装系统

密码错误



密码正确点击确定后即可切换到主界面。



<p>技术总结与问题分析</p>	<p>(给出详细的已完功能测试模块的技术总结,并对存在技术问题给出具体情境、界面、输入数据、输出结果等,并具体分析)</p> <p>1、计算机硬件系统</p> <p>应统一采用 UTF-16 的字符译码机制,自己新建的 Language 类效果并不是很好。</p> <p>2、文件管理</p> <p>没有实现新建文件,删除文件,文件共享的功能。文件保护过于简陋,仅仅通过开机时设置密码和设置 RWX 标志位是不够的。同时打开的文件有限制,一个进程只能同时打开一个文件。</p> <p>进程文件打开表和系统文件打开表在 GUI 显示信息可读性差。</p> <p>3、磁盘管理</p> <p>由于没有新建文件,删除文件的功能,所以无法对成组链接法的分配和回收磁盘块进行测试(代码中写了该方法)。</p> <p>4、内存管理</p> <p>固定分配会是的短作业资源浪费,并且 Inode 数目和缓冲区大小不匹配。</p> <p>5、作业管理</p> <p>静态优先级+先来后到算法还有有一定局限。</p> <p>6、进程管理</p> <p>五态模型的性能并不如七态模型。</p> <p>仅有一定预防死锁的功能,无法检测死锁并对其处理。</p> <p>进程信息在 GUI 界面显示信息可读性差。</p> <p>7、设备管理</p> <p>没有实现 Spooling 技术,仅仅采用最原始的中断管理。</p>
------------------	---

--	--