

内存管理项目



- 项目目的
- 开发环境
- 项目需求
- 项目提交
- 评核方式

同濟大學軟件學院





项目目的



- 动态分区分配方式
 - 数据结构、分配算法
 - 加深动态分区存储管理方式及其实现过程的理解。
- 请求分页分配方式
 - 页面、页表、地址转换
 - 页面置换过程
 - 加深对请求调页系统的原理和实现过程的理解。

同濟大學軟件學院





开发环境



- 自选操作系统平台
- 自选开发研究（如Java、vc）

同濟大學軟件學院



基本需求 (1/6)



● 动态分区分配方式的模拟

➤ 基本任务

假设初始态下，可用内存空间为640K，并有下列请求序列，请分别用首次适应算法和最佳适应算法进程内存块的分配和回收，并显示出每次分配和回收后的空闲分区链的情况来。

同濟大學軟件學院



项目需求 (2/6)



● 请求调页存储管理方式模拟

➤ 基本任务

假设每个页面可存放10条指令，分配给一个作业的内存块为4。模拟一个作业的执行过程，该作业有320条指令，即它的地址空间为32页，目前所有页还没有调入内存。



项目需求 (3/6)



模拟过程

- 在模拟过程中，如果所访问指令在内存中，则显示其物理地址，并转到下一条指令；如果没有在内存中，则发生缺页，此时需要记录缺页次数，并将其调入内存。如果4个内存块中已装入作业，则需进行页面置换。
- 所有320条指令执行完成后，计算并显示作业执行过程中发生的缺页率。

同濟大學軟件學院



项目需求 (4/6)



- 置换算法可以选用FIFO或者LRU算法
- 作业中指令访问次序可以按照下面原则形成：
50%的指令是顺序执行的，25%是均匀分布在前地址部分，25%是均匀分布在后地址部分。

数组中一个元素表示一个页中一条的指令
内存中显示出调出来的页，比较后决定调换与否



项目需求 (5/6)



➤ 具体实施方法如下（可以参考，不一定必须如此）

- 在0 – 319条指令之间，随机选取一个起始执行指令，如序号为 m
- 顺序执行下一条指令，即序号为 $m+1$ 的指令
- 通过随机数，跳转到前地址部分 $0 - m-1$ 中的某个指令处，其序号为 $m1$
- 顺序执行下一条指令，即序号为 $m1+1$ 的指令

同濟大學軟件學院



项目需求 (6/6)



- 通过随机数，跳转到后地址部分 $m1+2\sim 319$ 中的某条指令处，其序号为 $m2$
- 顺序执行下一条指令，即 $m2+1$ 处的指令。
- 重复跳转到前地址部分、顺序执行、跳转到后地址部分、顺序执行的过程，直到执行完320条指令。



项目提交



- 提交时间（12月10日前）
- 提交内容
 - 设计方案（word文档）
 - 源代码
 - 执行程序
- 提交方式
 - 学院服务器

同濟大學軟件學院



评核方式



- 考核点
 - 实现?
 - 界面?
 - 算法?
 - 可行性?

同濟大學軟件學院

