

实验报告

实验名称: 内存分配与分段分页机制

实验时间: 2018/4/23

实验人员: 李子强 (姓名) 11510352 (学号) 15 (年级)

实验目的: 掌握内存分配的过程, 了解碎片产生的原因和减少碎片的措施。理解分段与分页机制的原理。

实验环境: Linux

实验步骤:

1. 读懂源程序

2. 修改源程序

实验陈述:

1、基础知识回顾:

1. Uniprogramming 的内容与弊端: 独一享有内存, 编程简单。缺点不够有效率和功能简单。

2. Multiprogramming 的内容与弊端: 多程序运行, 操作系统处理更复杂, 需要考虑隔离, 内存翻译等问题, 使用过程中会不断产生碎片。

3. 分段机制的原理与利弊: 分段机制就是把虚拟地址空间中的虚拟内存组织成一些长度可变的称为段的内存块单元。优点, 便于翻译和隔离; 缺点, 段大小不确定, 可能会产生碎片。

4. 分页机制的原理与利弊: 分段机制的基础上完成了虚拟(逻辑)地址到物理地址转换的过程。分段机制把逻辑地址转换成线性地址, 而分页机制则把线性地址转换成物理地址。优点, 不产生碎片; 缺点, 需要额外的存储空间放置页表。

2、内存管理

内存块是通过哪种数据结构进行存储的? 链表

分配内存时如何减少内碎片的出现? 有多种分配内存算法, FF, BF, WF, 同时设定阈值, 在分配时不产生小于阈值的块。

简述理由 FF 尽量使用低地址, BF 尽量使用刚好的块, WF 尽量使用最大的块。从不同方面使得内存利用效率变高。

原始程序采取了怎样的策略进行内存分配? 分配低地址的可用内存, 如果分配剩余内存小于阈值则一起分配。

简述这样做的好处 循环使用低地址内存, 该算法倾向于使用内存中低地址部分的空闲区, 在高地址部分的空闲区很少被利用, 从而保留了高地址部分的大空闲区。显然为以后到达的大作业分配大的内存空间创造了条件。

3、遇到的问题与解决方法

问题 1: 段错误

解决方法 读取到 NULL

问题 2: scanf 读取的到异常数字

解决方法 地址先进行初始化

实验总结:

学习到了内存管理算法

```
1) Set memory size (default = 1024)
2) Set memory allocation algorithm
3) Create a new process
4) Kill a process
5) Display memory usage
233) Exit
1
Total memory size =2048

*****menu*****
1) Set memory size (default = 1024)
2) Set memory allocation algorithm
3) Create a new process
4) Kill a process
5) Display memory usage
233) Exit
5
*****Free Memory*****
      start_addr      size
              0      2048
Totally 1 free blocks
```

重新设定大小

```
*****Used Memory*****
      PID      ProcessName start_addr      size
      1      PROCESS-01      0      555
      2      PROCESS-02      555      443
      3      PROCESS-03      998      999
Totally 3 allocated blocks
```

显示已用内存

```
*****menu*****
1) Set memory size (default = 1024)
2) Set memory allocation algorithm
3) Create a new process
4) Kill a process
5) Display memory usage
233) Exit
3
Memory for PROCESS-04:222
Rearrange begins...
Rearrange by address...
Rearrange Done.
Allocation fail
```

分配过大内存失败

```
*****menu*****
1) Set memory size (default = 1024)
2) Set memory allocation algorithm
3) Create a new process
4) Kill a process
5) Display memory usage
233) Exit
2
    1) First Fit
    2) Best Fit
    3) Worst Fit
```

改变分配算法