



中山大學

SUN YAT-SEN UNIVERSITY

## 实 验 报 告

课程名称：\_\_\_\_ 操作系统 \_\_\_\_

姓 名：\_\_\_\_ 孙广岩 \_\_\_\_

学 号：\_\_\_\_ 20354242 \_\_\_\_

专业班级：\_\_\_\_ 智能科学与技术专业 5 班 \_\_\_\_

任课教师：\_\_\_\_ 吴贺俊 \_\_\_\_

\_\_\_\_ 2022 \_\_\_\_ 年 \_\_\_\_ 11 \_\_\_\_ 月 \_\_\_\_ 11 \_\_\_\_ 日

## 实验报告成绩评定表

[illegible]

## 实验 6.1 内存管理—动态分区

### 一、实验目的

1. 掌握动态分区分配方式使用的数据结构和分配算法
2. 进一步加深对动态分区分配管理方式及其实现过程的理解

### 二、实验内容

#### 1. 任务描述

补充 C 语言程序，模拟实现首次/最佳/最坏适应算法(样例代码使用的是首次适应算法、感兴趣同学可自行选择其它两种算法实现)的内存块分配和回收，要求每次分配和回收后显示出空闲分区和已分配分区的情况。假设初始状态下，可用的内存空间为 640KB。

#### 2. 实验方案

实验代码为 `dynamic_mem.c`。假设下列作业请求序列：

- (1) 作业 1 申请 130 KB
- (2) 作业 2 申请 60 KB
- (3) 作业 3 申请 100 KB
- (4) 作业 2 释放 60 KB
- (5) 作业 3 释放 100 KB
- (6) 作业 1 释放 130 KB

### 三、实验记录

首先是 firstfit 算法：

```
(base) PS C:\Files\SYSU> & C:/Apps/anaconda3/python.exe c:/Files/SYSU/dynamic_mem.py
首次适应法：1
最佳适应法：2
退出：其他任意键
请输入：1
* 申请130K
used: (0, 130)
not used: (130, 510)
=====
* 申请60K
used: (0, 130) (130, 60)
not used: (190, 450)
=====
* 申请100K
used: (0, 130) (130, 60) (190, 100)
not used: (290, 350)
=====
# 释放60K
used: (0, 130) (190, 100)
not used: (130, 60) (290, 350)
=====
# 释放100K
used: (0, 130)
not used: (130, 510)
=====
# 释放130K
used:
not used: (0, 640)
=====
```

请给

之后是 Bestfit 算法：

```
首次适应法： 1
最佳适应法： 2
退出： exit
请输入： 2
* 申请130K
used: (0, 130)
not used: (130, 510)
=====
* 申请60K
used: (0, 130) (130, 60)
not used: (190, 450)
=====
* 申请100K
used: (0, 130) (130, 60) (190, 100)
not used: (290, 350)
=====
# 释放60K
used: (0, 130) (190, 100)
not used: (130, 60) (290, 350)
=====
# 释放100K
used: (0, 130)
not used: (130, 510)
=====
# 释放130K
used:
not used: (0, 640)
=====
```

可以看到两个算法都成功的进行了内存的分配和回收。

## 实验 6.2 虚拟内存一页置换算法

### 一、实验目的

1. 掌握虚拟内存页分配方式使用的数据结构和页置换算法。
2. 进一步加深对虚拟内存和页置换算法实现的理解。

### 二、实验内容

1. 实现 LRU 算法，页容量自定，输入一段页序列，计算不同页容量下的页命中率；
2. 拓展：实现 FIFO 算法，并且分析 Belady 现象。

### 三、试验记录

实验采用了与理论课作业相同的设定，即页容量为 3 或者 4，页序列为 4 3 2 1 4 3 5 4 3 2 1 5 （可直接复制输入程序）

#### 1. LRU 算法

页容量为 3:

选择算法，LRU算法输入L，FIFO算法输入F: L				
页容量: 3				
页序列: 4 3 2 1 4 3 5 4 3 2 1 5				
Stream	Frame	→	0 1 2	Fault
4			4	Yes
3			4 3	Yes
2			4 3 2	Yes
1			1 3 2	Yes
4			1 4 2	Yes
3			1 4 3	Yes
5			5 4 3	Yes
4			5 4 3	No
3			5 4 3	No
2			2 4 3	Yes
1			2 1 3	Yes
5			2 1 5	Yes
页序列数: 12				
未命中数量: 10				
缺页率: 83.33%				

页容量为 4:

选择算法，LRU算法输入L，FIFO算法输入F: L				
页容量: 4				
页序列: 4 3 2 1 4 3 5 4 3 2 1 5				
Stream	Frame	→	0 1 2 3	Fault
4			4	Yes
3			4 3	Yes
2			4 3 2	Yes
1			4 3 2 1	Yes
4			4 3 2 1	No
3			4 3 2 1	No
5			4 3 5 1	Yes
4			4 3 5 1	No
3			4 3 5 1	No
2			4 3 5 2	Yes
1			4 3 1 2	Yes
5			5 3 1 2	Yes
页序列数: 12				
未命中数量: 8				
缺页率: 66.67%				

可以看到页容量的增加使 LRU 的缺页率下降。

#### 2. FIFO 算法

页容量为 3:

选择算法，LRU算法输入L，FIFO算法输入F: F				
页容量: 3				
页序列: 4 3 2 1 4 3 5 4 3 2 1 5				
Stream	Frame	→	0 1 2	Fault
4			4	Yes
3			4 3	Yes
2			4 3 2	Yes
1			1 3 2	Yes
4			1 4 2	Yes
3			1 4 3	Yes
5			5 4 3	Yes
4			5 4 3	No
3			5 4 3	No
2			5 2 3	Yes
1			5 2 1	Yes
5			5 2 1	No
页序列数: 12				
未命中数量: 9				
缺页率: 75.00%				

页容量为 4:

选择算法，LRU算法输入L，FIFO算法输入F: F				
页容量: 4				
页序列: 4 3 2 1 4 3 5 4 3 2 1 5				
Stream	Frame	→	0 1 2 3	Fault
4			4	Yes
3			4 3	Yes
2			4 3 2	Yes
1			4 3 2 1	Yes
4			4 3 2 1	No
3			4 3 2 1	No
5			5 3 2 1	Yes
4			5 4 2 1	Yes
3			5 4 3 1	Yes
2			5 4 3 2	Yes
1			1 4 3 2	Yes
5			1 5 3 2	Yes
页序列数: 12				
未命中数量: 10				
缺页率: 83.33%				

可以看到页容量的增加反而导致了缺页率的上升，出现了 Belady 现象。

### 四、心得体会

由于这两次实验原理上不是很复杂，所以使用了 Python 进行实现，然后还是更加深刻的理解了 Firstfit,Bestfit 算法，也更加熟悉了 LRU 和 FIFO 算法的原理。