



同濟大學
TONGJI UNIVERSITY

内存管理项目——请求分页分配方式设计方案

姓 名：叶栩冰

学 号：1953348

所 在 院 系：软件学院

任 课 教 师：张慧娟

二〇二一年六月

题目概述

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。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。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。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 $m1+1$ 的指令设计项目要包含界面、算法、可行性等等。

知识点概述

本项目为模拟请求分页分配方式，是虚拟内存管理的重要知识部分，请求分页系统建立于分有人系统基础之上，增加了请求调页功能和页面置换功能，是目前最常用的实现虚拟存储器的方法。主要核心部分为模拟存储空间、设计页表机制、模拟终端机构与模拟地址变换。

开发环境

由于要对请求分页分配方式进行模拟，贴近模拟操作系统架构，并且此项目用户交互功能较少，且要向用户展示的指令与内容信息过多，因此选择 c++ 语言进行相关开发模拟，利用 c++ 基本图形库进行被调入内存的页号显示，利用基本 menu 实现简单的交互选择，简单有效的向测试者展示指令的执行过程。

项目架构设计

本项目在设计时将整体程序架构设计为三个主要部分，分别为 **menu** 用户简单交互部分，请求调页算法模拟部分以及简单的图形化展示部分。由于整体逻辑较为简单，具体架构仅介绍算法模拟部分，详见后续的算法设计简介。

由于项目说明未明确规定计算机编址形式、地址初始条件与逻辑地址结构等详细信息，因此在此简要说明项目设计的计算机环境背景：

假设模拟请求分页分配的计算机按字节编址，且逻辑地址和物理地址都为 32 位，假设页表项大小为 4B，采用一级页表的分页存储管理方式。规定逻辑地址结构为 20 位页号与 12 位页内偏移量。假定被调用请求分配的内存空间的起始地址为 0080 0000H，由于页内偏移量为 12 位，转换为十六进制则为 3 位，因此页表内页框号从 00800 开始编号。

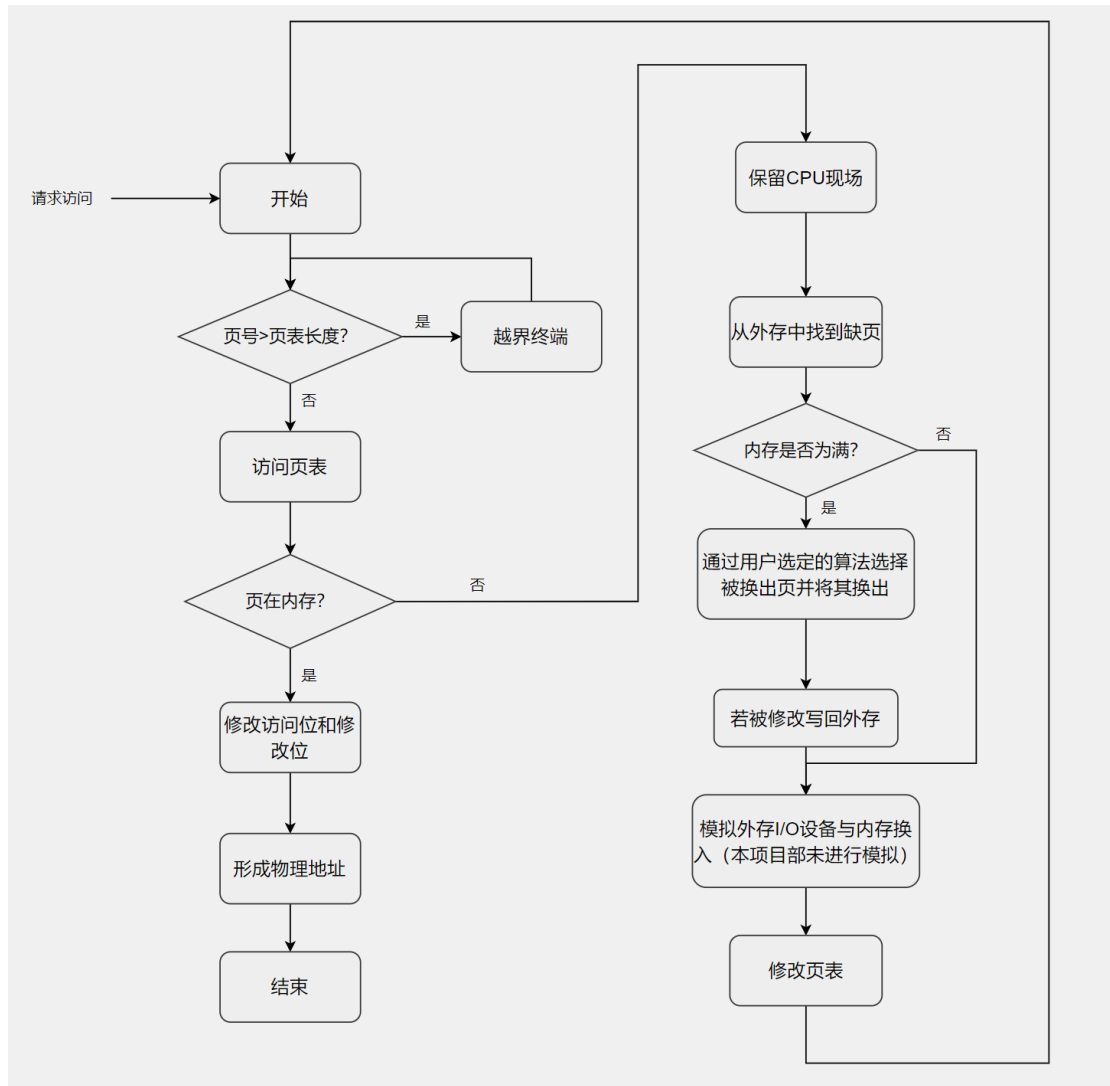
以上为项目设计的计算机环境背景假设，以下为简要的页表项结构体设计：

```
struct block
{
    int page_num = 0;
    string block_num = "";
    int status = 0; //0 不在 1 在
    string location_out = "";
};
```

其中 **page_num** 代表页号，**block_num** 代表物理块号即页框号，**status** 为状态位，**location_out** 代表指令的外存地址，由于本项目不存在对该页的修改以及访问字段，因此不设修改项。

请求分页分配设计

本项目的主要逻辑部分即为模拟请求分页分配，下图为本模拟方式单次运行的算法设计流程图：



其中核心部分为分页分配模块与页的置换模块，本项目利用 `visit` 函数对这两模块进行了模拟。首先会对页表项的状态位进行访问，来判断该页是否在内存中，若在内存中则根据模拟的起始地址与逻辑地址中的页内偏移量模拟生成实际的物理地址，进行输出提示，并对不同置换算法的访问队列进行维护，例如对于 `lru` 置换算法，要对返回时间与访问页以及最长最久未被访问的页号进行更新。若访问页不在内存中，首先依旧会修改访问位与状态位，然后就要去访问内存中的现有的内存块，判断内存是否为满，若未满则直接将页号加到各列表内，同时进行个数组的维护操作；若内存已满，则调用 `exchange()` 函数，该函数根据用户选择的不同算法对其进行了页面置换操作。

`Exchange()` 函数对置换函数进行了模拟，由于几种典型的置换函数课堂上有详细讲解，在此文档中不一一赘述。

由于程序提供三种不同的执行方式，因此在用户选择模拟不同执行方式后会输出缺页率。

UI 设计

```
=====
                          请选择指令执行方式
=====
A. 逐步执行指令(320条)
B. 一次性执行320条指令
C. 一次性执行该作业的全部指令
D. 退出模拟程序
=====
请输入执行方式：
```

由于不同的效果需求，本程序提供三种执行指令方式，分别为逐步执行，一次性执行 320 条，以及执行完成全部作业指令。

```
=====
                          请选择页面置换算法
=====
1. FIFO先进先出页面置换算法
2. LRU最近最久未使用算法
=====
请输入执行方式：_
```

在选择完执行方式后，可对置换算法进行选择。

```
*****
*   假定计算机主存按字节编址                               *
*   逻辑地址结构：页号为20位，页内偏移量为12位           *
*   块号即页框号为五位16进制，页表项大小为4B             *
*   设起始物理地址为0080 0000H                             *
*****

      3      20      10      15

正在执行第155条指令 指令所在页号为15 对应块号为 0080F
访问指令位置：不在内存内，缺页！
页号为23的页被调出内存, 页号为15的页被调入内存！
在内存中的页的页号为：3 20 10 15

按回车执行下一条指令...(输入q终止测试) _
```

此图为逐步执行指令的界面，根据页是否在内存内做出不同提示以及置换信息、地址信息、块号信息等。

```
当前访问页面失败次数为：149
当前执行指令条数为：320
缺页率为：0.465625

按回车返回菜单，输入q终止模拟程序 _
```

```
当前访问页面失败次数为：1045
当前执行指令条数为：2171
缺页率为：0.481345

按回车返回菜单，输入q终止模拟程序 _
```

选择一次性执行将一次性输出所执行指令的全部信息以及执行结果与缺页率。

详细测试见 exe 文件。