题目详见下边

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | | 备注 | 已考上机题 |  |
| 章3 | 可变分区存储管理方案中的**内存的分配**  关于动态分区算法: 最坏适应算法 首先适应算法 最佳适应算法  3\_1题：  1．程序运行时首先接收输入：空闲区数据文件，包括若干行，每行有两个数据项：起始地址、长度（均为整数），各数据项以逗号隔开。  2．建立**空闲区表**并在屏幕上显示输出空闲区表内容，空闲区表中记录了内存中可供分配的空闲区的始址和长度，用标志位指出该分区是否是未分配的空闲区。  3．从用户界面根据用户提示接收一个**内存申请**，格式为：作业名、申请空间的大小。  4．按照最差（最坏）适配算法选择一个空闲区，分割并分配，修改相应的数据结构（空闲区表），填写内存已分配区表（起始地址、长度、标志位），其中标志位的一个作用是指出该区域分配给哪个作业。  5．重复3、4，直到输入为特殊字符（0）。  6．在屏幕上显示输出新的空闲区表和已分配区表的内容。  本程序包括: 最坏适应算法 首先适应算法 最佳适应算法 | | 已练 | （网友miss考过的回忆版,**题不准，只能知大概考点**）  2012.11.14题：  设计两个表，一个表结构体有首块号和长度  第二个表是文件，有文件名 长度 占用的第一块号码和块数,内存连续存储  然后这两个表的基础上实现最先适应算法和最坏适应算法 | 过 |
| 侧重考察算法（3个） | |  |  |  |
| 3\_1\_2可变分区存储管理方案中的**内存回收**  用户提出内存空间的申请；系统根据申请者的要求，按照一定的分配策略分析内存空间的使用情况，找出能满足请求的空闲区，分给申请者；当程序执行完毕或主动归还内存资源时，系统要收回它所占用的内存空间或它归还的部分内存空间。  1．程序运行时首先接收输入：空闲区数据文件，包括若干行，每行有两个数据项：起始地址、长度（均为整数），各数据项以逗号隔开。  2．建立**空闲区表**并在屏幕上显示输出空闲区表内容，空闲区表中记录了内存中可供分配的空闲区的始址和长度，用标志位指出该分区是否是未分配的空闲区。  3．从用户界面根据用户提示接收一个内存回收申请，格式为：起始地址、长度（均为整数）。（用空格隔开）  4．按照可变分区**内存回收算法**修改空闲区表，要考虑四种可能性：  归还区有下邻空闲区；归还区有上邻空闲区；归还区既有上邻空闲区又有下邻空闲区；归还区既无上邻又无下邻空闲区。  5．重复3、4，直到输入为特殊字符（0）.  6．在屏幕上显示输出新的空闲区表  7．测试说明：测试教师将事先准备好一个文件，启动程序运行后，程序首先提示输入文件名，测试教师输入文件名，屏幕上首先显示空闲区表的内容，然后，测试教师输入其他数据，并查看程序输出结果 | | 已练 | 1. 会， 2. **基于链表的 简单回收**   2003.9.4  题目：内存分配和回收  1完成首次适配的动态内存分配  2完成动态内存的回收，只考虑释放一块内存后将它插入到空闲块链表的适当位置。  (2即找到位置，插入该节点即可；这只考书上的内存回收中最简单的一种) |  |
|  | |  |  |  |
|  | **3\_2\_1 分页管理(算法:**  1． 输入：页面流文件，其中存储的是一系列页面号（页面号用整数表示，用空格作为分隔符），用来模拟待换入的页面。下面是一个示意： 1 2 3 4 1 2 5 1 2 3 4 5  2． 处理要求：  程序运行时，首先提示"请输入页面流文件的文件名："，输入一个文件名后，程序将读入该文件中的有关数据。  初始条件：采用三个页框，初始时均为空。根据第二次机会算法对数据进行处理。  3． 输出要求：  每换入一个页面（即：每读入一个页面号），判断是否有页面需要被换出。若有，把被换出的页面号输出到屏幕上；若没有，则输出一个"\*"号。  4． 文件名约定  提交的源程序名字：sourceXXX.c或者sourceXXX.cpp（依据所用语言确定）  输入文件名字：可由用户指定 其中：XXX为账号。  5． 测试说明：测试教师将事先准备好一组文件（格式为\*.txt），从中为每个程序随机指定一至三个作为输入文件（被测试者需从键盘输入指定文件的文件名）并查看程序输出结果。  (3\_2\_1)6． 第二次机会算法：对FIFO算法做如下简单的修改：发生替换时，先检查最老页面的R（访问）位。如果为0，那么此页面是最早被换入的，而且近期没有被访问，可以立刻被替换掉；如果R位为1，就清除R位，并修改它的装入时间，使它就像刚被装入的新页面一样，然后继续搜索可替换的最老页面。 | |  | 如2001年考题： 要求：  1。实现三种算法： FIFO，最近最少使用调度算法（LRU），最近最不常用调度算法（LFU）  2。页面序列从指定的文本文件（TXT文件）中取出  3。输出： 第一行：每次淘汰的页面号  第二行：显示缺页的总次数 |  |
|  | **3\_2\_2页式内存的分配与回收 (3道)**  **见word:** **A1 总体安排的附件3\_2\_1** | |  |  |  |
| 四  文件  +磁盘 | 4\_1文件操作  （2年真题） | |  | （真2001年）  建立一个树型文件目录  假设程序启动运行后在根目录下且根目录为空。  实习检查：   1. 运行程序，由检查教师给出文件名，该文件中存有相应的若干命令。   （程序应做提示，界面友好）。  2、要求实现两个命令：  mkdir 目录名（目录已存在，应给出错误信息。）  cd 目录名（目录不存在，应给出错误信息。）  3、你所编制的程序应读入文件，并执行其中的每一条命令。  4、在屏幕上显示文件目录的结构。（界面自己设计，但要清晰明了。） |  |
| 真2000.9 (已练,花了（ 4天的时间解决的  题目：模拟文件系统  题见：题4\_1\_2 模拟文件系统2000.9.doc |
| 4\_2(磁盘调度 :4种 见算法  三，模拟磁盘驱动调度过程：  　　20个磁头（0-19） 共有200个柱面；  　　由外向内（0-199）  　　程序运行：  　　（1），老师给出的文件名，文件中存有磁盘访问请求的柱面号。  　　（2），按照先来先服务磁盘调度算法，在题目上显示调度结果。（磁盘访问请求的磁道号以及磁头移动的总磁道数）  　　（3），假设当前磁头的位置在57磁道上按照最短寻道时间优先磁道访问算法，在屏幕上显示调度结果以及磁头移动的总磁道数。 | | 已练 | 2001年的试题 :磁盘调度算法  要求：  1。实现三种算法：1。先来先服务 2。最短寻道优先（老师会给当前磁头的位置） 3。电梯算法  2。磁道服务顺序从指定的文本文件（TXT文件）中取出  3。输出： 第一行：磁道的服务顺序 第二行：显示移动总道数 | 已练 |
| 4\_3 文件与位示图 | |  | 真2002年的  内存被划分成2048块（页）。用32位字长的字存放位示图，为0的位表示该块尚未分配，为1的位表示该块已分配  实习检查：  1、运行程序，由检查教师给出文件名，该文件中存有内存目前状况的位示图的数据（0和1的文件）。（程序应做提示，界面友好）。  2、你所编制的程序应读入数据，存放在相应的数据结构中。  3、显示友好的用户界面，由检查教师输入内存申请（总块数）。  4、根据申请和位示图状态，为用户分配内存，并建立页表。  5、输出位示图和页表。 | 已练 |
| 章二 |  | |  | 真题考知识点的某个细节  （如2007 考优先数调度； 某年6月考高响应比调度算法 |  |
| 2\_22: 进程调度(先不做  进程调度算法有FIFO，优先数调度算法，时间片轮转调度算法，分级调度算法，目前主要是考FIFO和优先数调度算法（静态优先级。  输入：进程流文件，其中存储的是一系列要执行的进程，每个作业包括四个数据项：  进程名 进程状态(1就绪 2等待 3运行) 所需时间 优先数(0级最高)  输出:  进程执行流 和 等待时间 平均等待时间 | |  | (2007年的) 2\_1进程的就绪队列  1．设计一个数据结构“进程控制块”  进程控制块PCB记录了进程的各种管理和控制信息。主要包括：进程号（系统赋予，正整数）、进程名、进程优先级（正整数）、进程状态、进程队列指针、其他信息（可以用一串字符表示）  **优先级越高，整数值越大**。  2．建立进程就绪队列  进程创建要求存放在一个.txt文件中。每行2个数（用空格隔开），一个是进程名，一个是进程优先级（正整数）。要求按照优先级高低建立进程就绪队列。  程序运行流程  1．提示用户“输入进程创建要求文件：”，读入一个.txt文件。  2．根据读入数据建立就绪队列：  （1）读一行数据，生成一个进程号，赋予该进程；  （2）生成一个空的进程控制块，根据有关信息填写其内容；  （3）将该进程控制块按照优先级高低链接到进程就绪队列中；  （4）重复（1）-（3），直到遇到特殊字符0。  提示：用一个指针指向进程就绪队列队首。  3．输出进程就绪队列（一行中的数字用逗号隔开）。  注意：1．代码中数据结构的设计要给出注释。  2．用户输入提示要清晰。  3．输出表格时格式要清晰 | 已练 |
|  | 2\_23作业调度  作业调度主要有FIFO，运算时间短的作业优先，优先数调度算法，响应比最高者优先调度算法，均衡调度算法  在一个多道程序设计系统中，实现"作业调度"的模拟。模拟程序的要求如下：  1．输入：  作业流文件，其中存储的是一系列待执行的作业信息，每个作业包括五个数据项：  作业号，优先级，进入输入井时间（用一个整数表示，如10：10表示为1010），  作业估计运行时间（单位：分钟），内存需求量。各数据项用空格隔开  2．输出：  按照作业调度次序输出每一个作业流文件："作业号"、"进入内存时间"、  "作业完成时间"、"周转时间"；每行输出一个作业的信息。  计算出平均周转时间并输出。  3．作业调度算法：  短作业优先（作业估计运行时间最短的作业首先被调度）。  4．处理要求：  在两道环境下有多个作业；  程序运行时，首先提示输入测试文件的文件名；  输入文件名后，程序将读入文件中的有关数据，并按照最短作业优先调度算法  对作业进行调度，作业被调度运行后不再退出（忽略进程调度或者可以认为  进程调度采用的是先来先服务调度算法）；  假设内存容量为100K，存储管理方案为可变分区方式，且不允许在内存中移动  作业。  数据结构定义：  struct job {  int job\_ID; //作业序  int run\_time; //估计运行时间  int memory; //内存需求量（单位K）  int sched; //调度标记，标记该作业在调度过程中是否被选中：0代表未被调度，1代表已调度  }; | |  | 2\_2 ( 6月四号上午的考题  模拟最高响应比优先作业调度算法  输入：作业流文件，其中存储的是一系列要执行的作业，  每个作业包括三个数据项：  作业号、作业进入系统的时间（用一整数表示，如10：10，表示成1010）、估计执行时间（单位分）优先级（0级最高）  参数用空格隔开，下面是示例：  1 800 50  2 815 30  3 830 25  4 835 20  5 845 15  6 900 10  7 920 5  其中调度时刻为最后一个作业到达系统的时间！  输出作业号！进入内存的时间！每行输出一个作业信息 | 已练 |
| 下面的先缓 | | | | |  |
|  | 2\_3  银行家算法  银行家可以把一定数量的资金供多个用户周转使用，为保证资金的安全银行家规定：  1.当一个用户对资金的最大需求量不超过银行家现有的资金就要接纳该用户;  2.用户可以分期贷款,但垡的总数不能超过最大需求量;  3.当银行家现有的资金不能满足用户的沿需垡数时,对用户的贷款可推迟支付,但总能使用户在有限的时间里得到贷款;  4当用户得到所需的全部资金后,一定能在有限的时间里归还所有的资金.  假设共有3类资源A B C,分别为10,5,7个  进程列表存在文件中,格式为  进程名 对A类资源需求数量 对B类资源需求数量 对c资源需求数量  输出每个进程的处理结果 |  | |  |  |
|  | 2\_4模拟死锁检测算法  1． 输入：  "资源分配表"文件，每一行包含资源编号、进程编号两项（均用整数表示，并用空格分隔开），记录资源分配给了哪个进程。  "进程等待表"文件，每一行包含进程编号、资源编号两项（均用整数表示，并用空格分隔开），记录进程正在等待哪个资源。  下面是一个示例：  资源分配表：  1 1  2 2  3 3  进程等待表：  1 2  2 3  3 1  2． 处理要求：  程序运行时，首先提示"请输入资源分配表文件的文件名："；再提示"请输入进程等待表文件的文件名："。  输入两个文件名后，程序将读入两个文件中的有关数据，并按照死锁检测算法进行检测。  3． 输出要求：  第一行输出检测结果：有死锁 或 无死锁。  第二行输出进程循环等待队列，即进程编号（如果有死锁）。  4． 文件名约定  提交的源程序名字：resourceXXX.c或者resourceXXX.cpp（依据所用语言确定）  输入文件名字：可由用户指定  结果输出到resultXXX.txt中  其中：XXX为账号。  5． 死锁检测算法：当任一进程Pj申请一个已被其他进程占用的资源ri时，进行死锁检测。检测算法通过反复查找进程等待表和资源分配表，  来确定进程Pj对资源ri的请求是否导致形成环路，若是，便确定出现死锁。  6． 测试说明：测试教师将事先准备好一组文件（格式为\*.txt），从中为每个程序随机指定一至三个作为输入文件  （被测试者需从键盘输入指定文件的文件名），并查看程序输出结果。 |  | |  |  |