

苏州大学 操作系统 课程试卷 (A) 卷 共 7 页

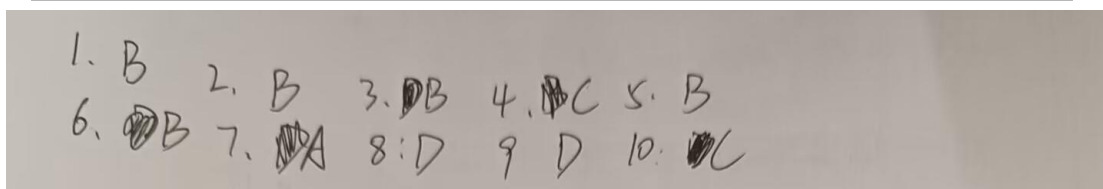
考试形式 开 卷 2022 年 12 月

院系 计算机 年级 2020 级 专业 人工智能

学号 2027407077 姓名 陈琦 成绩

一、 选择题 (20 分, 每题 2 分, 答案请填入下表, 否则不等分!)

1	2	3	4	5	6	7	8	9	10



1、分页式虚拟存储系统中, 页面的大小与可能产生的缺页中断次数_____。

- (A) 成正比 (B) 成反比
(C) 成固定比例 (D) 无关

2、页式管理系统中, 地址寄存器的低 9 位表示页内地址, 则页面大小为_____。

- (A) 1024B (B) 512B (C) 1024KB (D) 512KB

3、某系统采用了银行家算法, 则下列说法正确的是_____

- (A) 系统处于不安全状态时一定会发生死锁
(B) 系统处于不安全状态时可能会发生死锁
(C) 系统处于安全状态时一定发生死锁
(D) 系统处于安全状态时可能会发生死锁

4、一种既有利于短作业又兼顾到长作业的作业调度算法是_____。

- (A) 先来先服务 (B) 时间片轮转

(C) 最高响应比优先 (D) 短作业优先

5、为了允许文件系统中存在具有相同的文件名，通常文件系统采用_____。

(A) 文件控制块 (B) 多级目录 (C) 索引结点 (D) 单级目录

6、一个目录包含 100 个文件，每个文件的目录项为 1.3KB，则这个目录文件需要_____个大小为 4KB 的物理块来存放。

(A) 100 (B) 33 (C) 32 (D) 34

7、逻辑文件的访问方式不包括_____。

(A) 间接访问 (B) 顺序访问 (C) 随机访问 (D) 直接访问

8、现代 OS 具有并发性和共享性，是_____的引入导致的。

(A) 单道程序 (B) 磁盘 (C) 对象 (D) 多道程序

9、设两个进程共用一个临界资源的互斥信号量 `mutex`，当 `mutex = 1` 时表示_____。

(A) 两个进程进入临界区 (B) 两个进程都在等待
(C) 一个进程进入了临界区，另一个进程等待 (D) 没有一个进程进入临界区

10、一个虚拟存储器中，段的逻辑地址形式是段号 10 位，段内地址 20 位；内存 1MB，辅存 10GB。那么该虚拟存储器最大实际容量可能是_____。

(A) 1024KB (B) 10GB (C) 1024MB (D) 10GB+1MB

二、简答题（20 分，每题 5 分）

1. 系统颠簸的原因是什么？一旦系统检测到颠簸，怎样才能消除这个问题？

1. 系统颠簸的原因: 分配的页数少于进程所需的最小页数, 时会发生系统颠簸

系统颠簸, 并迫使它不断地发生缺页错误。系统可以通过对比多道程序的程度来估计CPU利用率的程度, 以此来检测颠簸。

通过局部置换算法或优先级置换算法可以限制系统抖动。系统可以给予进程更多的页框数或是降低系统的多道程度来消除颠簸现象。

2. 论述长期、中期、短期调度之间的区别。

2. (1) 短期调度: 从就绪可执行的进程中选择进程并为其之一分配CPU
(2) 中期调度: 特别用于分时系统中作为中等程度调度程序。能将进程移出内存, 并移出对CPU的激烈竞争, 因此降低多道程序设计的程度。之后进程能被重新调入内存, 并从中断处继续执行
(3) 长期调度: 从缓冲池中选择进程, 并将它们装入内存以执行。
它们的主要区别是执行的频率。短期调度要频繁地为CPU选择新的进程执行。长期调度执行的频率较低, 用于控制多道程序设计的程度, 即内存中的进程数量。中期调度介于两者之间。
短期调度频率较大, 中期调度频率较小, 长期调度频率较大。

3. 试述内碎片与外碎片的区别。

3. 性质不同:

外碎片: 是还没有被分配出去, 即不属于任何进程, 但由于太小无法分配给申请内存空间的新进程的内存空闲区域

内碎片: 是已经被分配出去, 即能明确指出其属于哪个进程, 却不能被利用的内存空间

存储块不同:

外碎片: 是处于任何已分配区域或页面外部的空闲存储块

内碎片: 是处于区域内部或页面内部的存储块

状态不同:

外碎片: 这些存储块的总和可以满足当前申请的长度要求, 但由于它们的地址不连续或其他原因, 使系统无法满足当前申请
内碎片: 占有这些区域或页面的进程, 并不使用这个存储块。而在进程占有这块存储块时, 系统无法利用它。直到进程释放它, 或进程结束时, 系统才有可能利用这个存储块。

4. 某一系统有 2 个进程和 3 个相同类型的资源。每个进程最多需要两个资源。这种情况下有没有可能发生死锁？为什么？

4. 不可能发生。假设每个进程都拥有一个资源。有一个资源是空闲的。任何一个进程可以要~~求~~请求它，并获得它，在这种情况下，它可以完成和释放这两种资源。因此，死锁是不会发生的。

三、 综合题（60 分，每题 10 分）

1、（10 分）一个分页存储系统，内存读/写一个单元的时间为 100ns，页表存放在内存：

- （1）如果采用单级页表，则访问一个内存地址存放的数据需要多少时间？（2 分）
- （2）如果采用三级页表，则访问一个内存地址存放的数据需要多少时间？（2 分）
- （3）如果一个单级页表分页系统引入联想寄存器 TLB，85%的页表项可在快表中命中，则访问一个内存地址存放的数据平均需要多少时间？（假设访问一次快表需要 5ns）。（2 分）
- （4）如果系统采用请求式分页（单层页表，不采用 TLB），缺页率为 0.1%，缺页处理总的时间为 10ms，则访问一个内存地址存放的数据平均需要多少时间？（2 分）

三

1. (1) $t_1 = 100 + 100 = 200 \text{ ns}$

(2) $t_2 = 100 \times 3 + 100 = 400 \text{ ns}$

(3) $t_3 = 0.85 \times 5 + 0.15 \times (5 + 2 \times 100) = 19.25 \text{ ns}$

(4) $t_4 = (1 - 0.001) \times 200 + 0.001 \times 10^4 = 209.8 \text{ ns}$

2、（10 分）在银行家算法中，如果出现以下资源分配情况

	Allocation	Need	Available
P0	0 0 3 2	0 0 1 2	1 6 2 2
P1	1 0 0 0	1 7 5 0	
P2	1 3 5 4	2 3 5 6	
P3	0 3 3 2	0 6 5 2	
P4	0 0 1 4	0 6 5 6	

- （1）该状态是否安全？请给出过程（6 分）
- （2）如果进程 P2 提出请求 Request(1,2,2,2)后，系统能否将资源分配给他？（4 分）

2,

解: (1) 利用银行家算法对此时刻进行分析, 可得安全性分析情况如下

	Work	Need	Allocation	Work+Allocation	Finish
P ₀	1 6 2 2	0 0 1 2	0 0 3 2	1 6 5 4	true
P ₃	1 6 5 4	0 6 5 2	0 3 3 2	1 9 8 6	true
P ₄	1 9 8 6	0 6 5 6	0 0 1 4	1 9 9 10	true
P ₁	1 9 9 10	1 7 5 0	1 0 0 0	2 9 9 10	true
P ₂	2 9 9 10	2 3 5 6	1 3 5 4	3 12 14 14	true

由上可得存在序列 $\langle P_0, P_3, P_4, P_1, P_2 \rangle$ 满足安全标准
即该状态是安全的

(2) P₂ 提出请求 Request (1, 2, 2, 2) 后, 按银行家算法进行检查:

Request (1, 2, 2, 2) \leq Need (2, 3, 5, 6)

Request (1, 2, 2, 2) \leq Available (1, 6, 2, 2)

试分配并修改数据结构, 资源分配如下

	Allocation	Need	Available
P ₀	0 0 3 2	0 0 1 2	0 4 0 0
P ₁	1 0 0 0	1 7 5 0	
P ₂	2 5 7 6	1 1 3 4	
P ₃	0 3 3 2	0 6 5 2	
P ₄	0 0 1 4	0 6 5 0	

再利用安全性算法检查发现 Available (0, 4, 0, 0) 已不能满足任何进程的需要, 故系统进入不安全状态,
即系统不能将资源分配给 P₂

3、(10 分) 假设现在有 5 块内存可以分配, 依次为 100KB, 500KB, 200KB, 300KB 和 600KB。有进程序列分别要求 212KB, 417KB, 112KB 和 426KB。

(1) 用首次适应, 最佳适应和最差适应分别能够按怎样的顺序分配进程? (9 分)

(2) 哪个算法充分利用了内存空间? (1 分)

3. (1) 首次适应:

212KB选中分区2, 此时分区2还剩128KB

417KB选中分区5, 此时分区5还剩183KB

112KB选中分区2, 此时分区2还剩176KB

426KB无分区能满足, 应该等待

最佳适应:

212KB选中分区4, 此时分区4还剩88KB

417KB选中分区2, 此时分区2还剩83KB

112KB选中分区3, 此时分区3还剩88KB

~~426KB~~选中分区5, 此时分区5还剩174KB

最差适应:

212KB选中分区5, 此时分区5还剩388KB

417KB选中分区2, 此时分区2还剩83KB

112KB选中分区5, 此时分区5还剩276KB

426KB无分区能满足, 应该等待

2) 最佳适应算法能最~~有效~~利用内存
充分

4、(10分) 在页式虚拟存储管理的计算机系统中, 作业在主存中分配到3块主存空间, 作业执行时访问页的顺序为 2,3,2,1,5,2,4,5,3,2,5,3, 请问用 OPT, FIFO 和 LRU 替换算法时, 它们的缺页中断率分别是多少。(要求图示出内存页面变化情况)

4. 1. OPT算法:

访问	2	3	2	1	5	2	4	5	3	2	5	3
内存空间	2	2 3		2 3 1	2 3 5		4 3 5		3 2 3 5			

$$\text{缺页率} = \frac{6}{12} = 50\%$$

2. FIFO算法:

访问	2	3	2	1	5	2	4	5	3	2	5	3
内存空间	<div>2</div>	<div>2 3</div>		<div>2 3 1</div>	<div>5 3 1</div>	<div>5 2 1</div>	<div>5 2 4</div>		<div>3 2 4</div>		<div>3 5 4</div>	

$$\text{缺页率} = \frac{8}{12} = 66.7\%$$

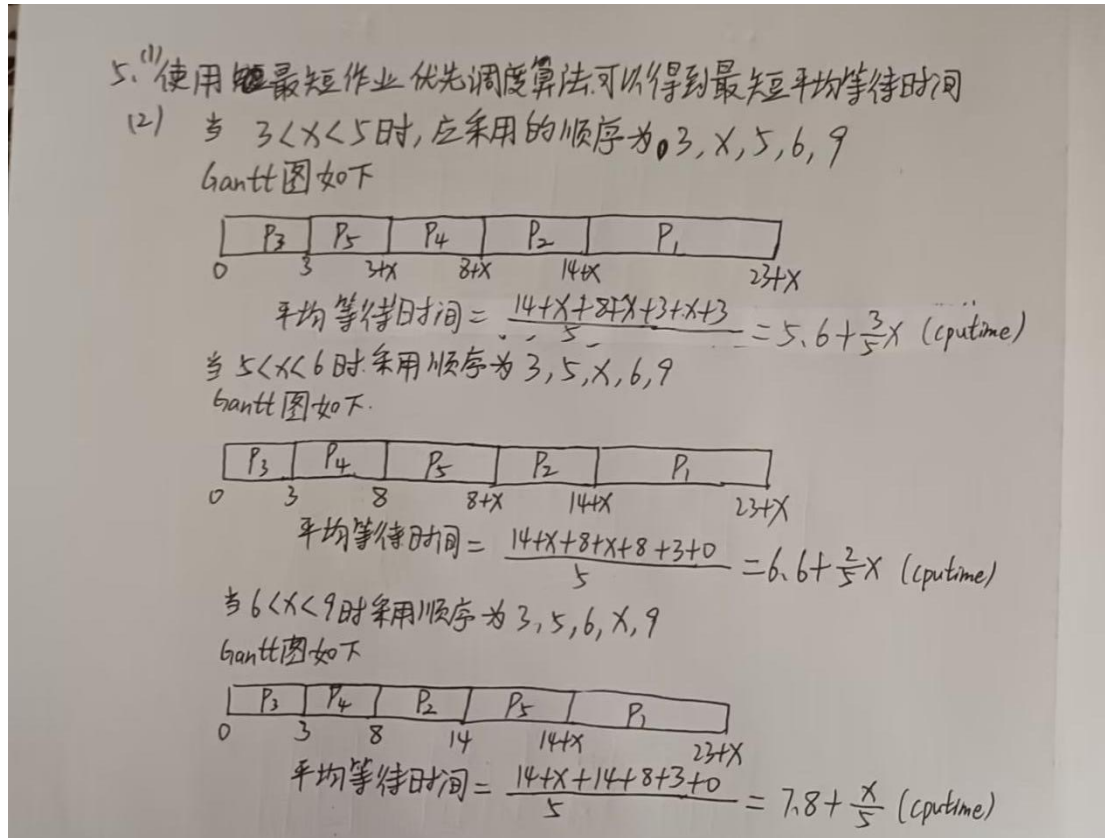
3. LRU算法:

访问	2	3	2	1	5	2	4	5	3	2	5	3
内存空间	<div>2</div>	<div>2 3</div>		<div>2 3 1</div>	<div>2 5 1</div>	<div>2 5 4</div>		<div>3 5 4</div>		<div>3 5 2</div>		

$$\text{缺页率} = \frac{7}{12} = 58.3\%$$

5、(10分) 有5个待运行进程同时到达，它们要运行的CPU脉冲时间N分别是9,6,3,5和X (3<X<9并且不等于5或6)。请问:

- (1) 采用哪种调度方法运行这5个进程可以得到最短的平均等待时间? (3分)
- (2) 按照以上调度方法, 请给出5个进程的运行次序, 并计算平均等待时间。 (7分)



6、(10 分) 下面是用 wait/signal 机制解决写者优先的 Readers-Writers Problem 的伪代码。所谓写者优先的 Readers-Writers Problem 定义如下:

- (1) 一块初始化已有内容的缓冲区, 由读者和写者共享;
- (2) 写者之间、写者和读者之间互斥使用缓冲区;
- (3) 读者之间可以同时访问缓冲区;
- (4) 当写者 i 需要写时: 1) 如果有写者 $i-1$ 正在写, 那么写者 i 等待, 且其优先于任何等待的读者; 2) 如果缓冲区空, 写者 i 使用缓冲区; 3) 如果缓冲区被若干读者使用, 不剥夺读者的缓冲区, 等到所有已经占用缓冲区的所有读者完成, 写者 i 立刻进入缓冲区;
- (5) 读者仅当没有写者正在或需要使用缓冲区时, 才进入缓冲区。

具体代码如下, 请在下面空格处填空:

int rcount, wcount; (initial value = _____)

semaphore m1, m2, m3, w, r; (initial value = 1)

READER:

wait(m1); //所有的 readers 竞争抢 w 信号量的权利

wait(r); //所有抢 readers 竞争 w 信号量的过程是互斥的

wait(m2);

rcount := _____;

if rcount = 1 then wait(_____); //只有第一个 reader 能够有资格抢 w 信号量

```

    _____ m2 _____;
    signal(r);
    signal(_____ m1 _____);
reading
    wait(m2);
    rcount := _____;
    if rcount = 0 then signal(_____); //还有 readers 在等，就不放 w 信号量
    signal(m2);

```

WRITER:

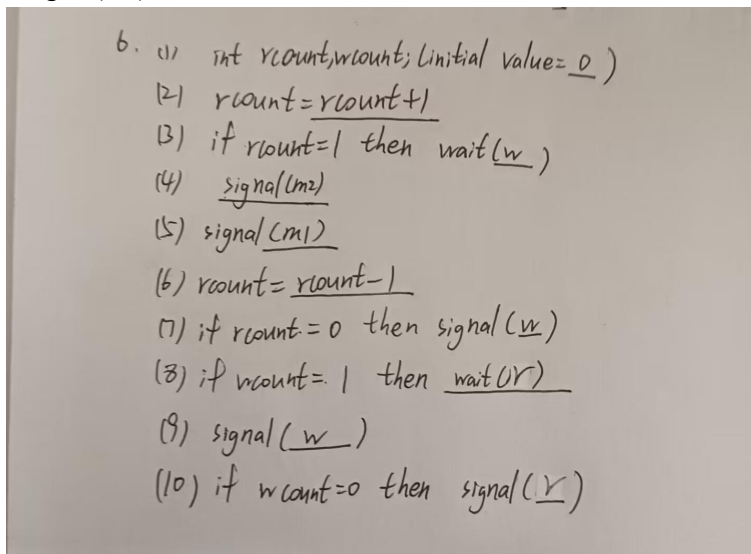
```

    wait(m3);
    wcount := wcount + 1;
    if wcount = 1 then _____;
    signal(m3);

    wait(w);
writing
    signal(_____);

    wait(m3);
    wcount := wcount - 1;
    if wcount = 0 then signal(_____);
    signal(m3);

```



6. (1) $\text{int } rcount, wcount; \text{ initial value} = 0$

(2) $rcount = rcount + 1$

(3) if $rcount = 1$ then wait(w)

(4) signal(m2)

(5) signal(m1)

(6) $rcount = rcount - 1$

(7) if $rcount = 0$ then signal(w)

(8) if $wcount = 1$ then wait(w)

(9) signal(w)

(10) if $wcount = 0$ then signal(w)