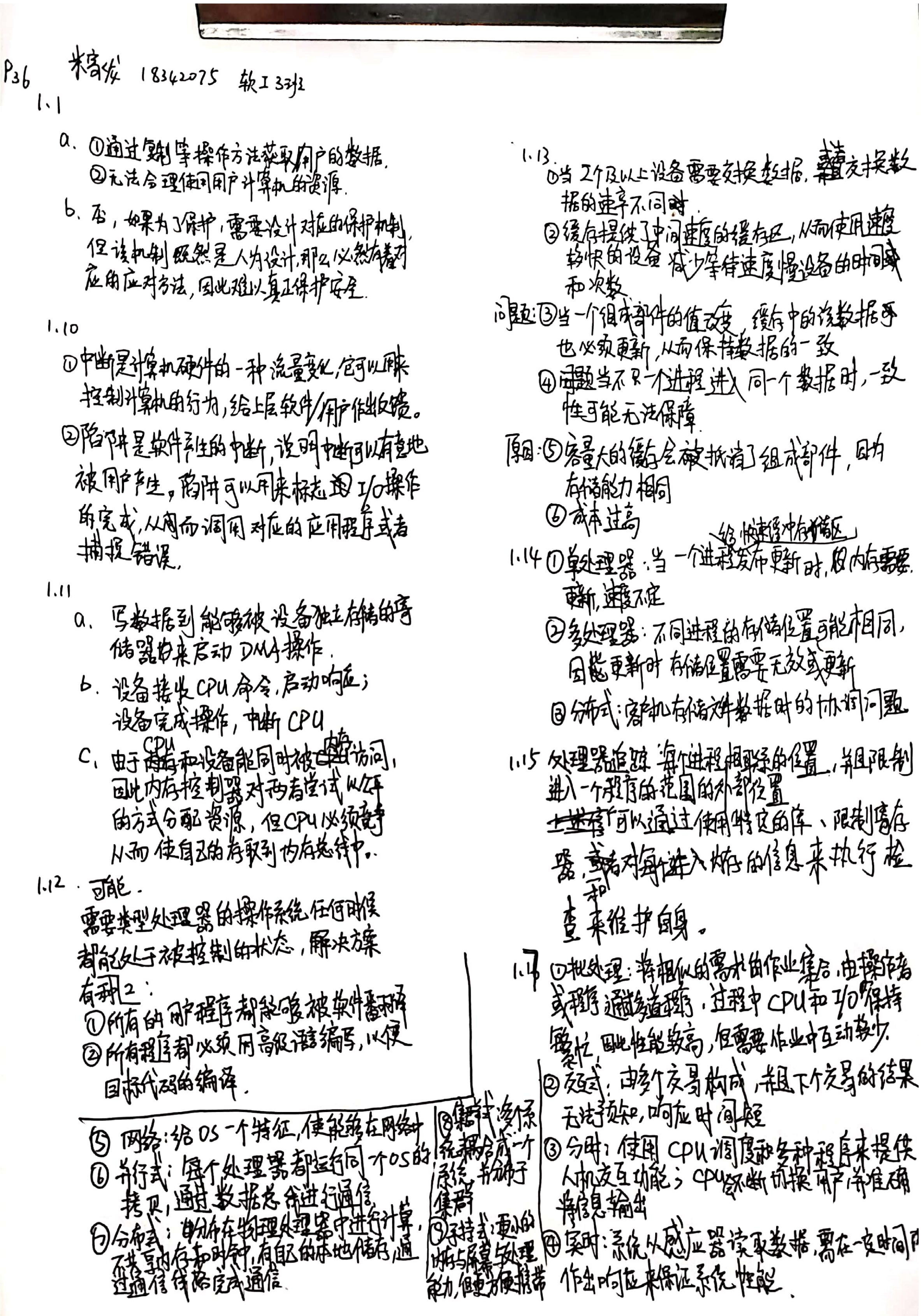
操作系统第一次书面作业

- 数据科学与计算机学院
- 软工三班
- 米家龙
- 18342075



P36 米家龙 18342075 软工3班

1.1

a. ①通过复制等操作方法获取用户的数据.
②无法合理使用用户计算机的资源.

b. 否，如果为了保护，需要设计对应的保护机制，但该机制既然是人为设计，那么必然存在对应的应对方法，因此难以真正保护安全.

1.10

①中断是计算机硬件的一种流量变化，它可以用来控制计算机的行为，给上层软件/用户作出反馈。

②陷阱是软件产生的中断，说明中断可以有意地被用户产生，陷阱可以用来标志I/O操作的完成，从而调用对应的应用程序或者捕捉错误.

1.11

a. 写数据到能够被设备独立存储的寄存器中来启动DMA操作.

b. 设备接收CPU命令，启动响应；设备完成操作，中断CPU

c. 由于CPU和设备能同时被内存访问，因此内存控制器对两者尝试以公平的方式分配资源，但CPU必须竞争从而使自己的存取到内存总线中。

1.12 可能.
需要该类型处理器的操作系统任何时候都能处于被控制的状态，解决方案有两种：
①所有的用户程序都能够被软件翻译
②所有程序都必须用高级语言编写，以便目标代码的编译.

1.13
①当2个及以上设备需要交换数据，或者交换数据的速率不同时.
②缓存提供了中间速度的缓存区，从而使用速度较快的设备减少等待速度慢设备的时间和次数
问题：③当一个组成部件的值改变，缓存中的该数据也必须更新，从而保持数据的一致
④当不只一个进程进入同一个数据时，一致性可能无法保障
原因：⑤容量大的缓存会被抵消了组成部件，因为存储能力相同
⑥成本过高

1.14 ①单处理器：当一个进程发布更新时，给快速缓冲存储区，内存需要更新，速度不定
②多处理器：不同进程的存储位置可能相同，因此更新时存储位置需要无效或更新
③分布式：客户机存储文件数据时的协调问题

1.15 处理器追踪每个进程相联系的位置，并且限制进入一个程序的范围的外部位置
可以通过使用特定的库、限制寄存器，和对每个进入内存的信息来执行检查来维护自身。

1.17 ①批处理：将相似的需求的作业集合，由操作者或程序通过多道程序，过程中CPU和I/O保持繁忙，因此性能较高，但需要作业中互动较少.
②交互式：由多个交易构成，并且下个交易的结果无法预知，响应时间短
③分时：使用CPU调度和多种程序来提供人机交互功能；CPU会不断切换用户并准确将信息输出
④实时：系统从感应器读取数据，需在一定时间内作出响应来保证系统性能.
⑤网络：给OS一个特征，使能够在网络中
⑥并行式：每个处理器都运行同一个OS的拷贝，通过数据总线进行通信
⑦分布式：单独存在物理处理器中进行计算，不共享内存和时钟，有自己的本地储存，通过通信线路完成通信
⑧集群式：将系统耦合成一个系统，并分布于集群
⑨手持式：更小的内存与屏幕与处理能力，但更方便携带

林宇松

林宇松 18342075 软工3班

8.1 内部碎片是在页或区域中~~未被占用该程~~被占用但未被使用的空间。只有当作业完成，或是并且该页/区域被释放后，系统才能使用。

8.2
① 将未解析符号地址替换为其与最终目标二进制文件中的变量相关联的实际地址。
② 需要跟踪引用了未解析符号的地址。

8.3
① First-fit
212K - 500K
417K - 600K
112K - (500-212)K = 288K
426K 需要等待

② Best-fit: 最优解
212K - 300K
417K - 500K
112K - 200K
426 - 600K.

③ Worst-fit
212K - 600K
417K - 500K
112K - ~~500~~ 388K
426K 须等待

8.4 ① 连续内存分配：当没有足够的空间供程序增加，需要重新分配整个程序。
② 分段 遇到上述的情况，也需要重新定位要扩展的分段。
③ 分纯页：不需要重新分配和定位其地址空间，只需增量分配新页面

8.5
连续分配：
① 连续分配内存会受到外部碎片的影响，并且自带漏洞，会随着新旧进程的轮替而不断增多
② 因为进程的虚拟内存段无法分成不连续的细粒度段，所以该方案不允许共享代码。
纯分段：
③ 由于在该方案下，分段在物理内存中是连续布置的，当旧进程被杀，新进程分段替换进入时，也会产生外部分段
④ 可以共享代码。
纯分页：
⑤ 不受外部碎片影响，但存在内部碎片
⑥ 进程按页面粒度分配，但页面如果未被充分利用则会导致内部碎片的出现和空间的浪费。

8.6 页系统的地址由逻辑页号和偏移量组成。可物理页面通过在逻辑页号表来找到物理页，并用操作系统可以将~~分配给进程的物理页 限制为仅访问~~设置进程为仅能访问分配到的物理页表，因此进程无法引用没被分配到的页面。
如果进程要访问，只能让OS将该组加入进程的物理页表中，从而保证安全。
当进程之间需要交换数据时，它们通过读取和写相同的物理地址来实现，通信时使用的是虚拟地址，从而保证通信的高效和安全。

8.7 ① 分页需要维护转换结构，因此要更多内存
② 分段需要2个寄存器，一个用于维护分段基础，另一个用于维护分段范围
③ 分页的每个分页都需要一个条目来存储物理地址。

8.8 ① 连续内存分配要求OS在程序开始前提分配一个足够大的虚拟内存的范围
② 纯分段可以在程序启动时为每个分段分配足够的范围，在需要更多的空间时可灵活扩展该分段
③ 纯页面调度不需OS在程序启动时分配虚拟地址空间的最大范围，但仍需要分配跨程序所有虚拟空间的大页表。当程序需扩展时，只需分配一个条目配设置好的新页即可

HUAWEI Mate 30 5G
SuperSensing Camera | LEICA

米家龙 18342075

8.9.

a. 200ns 用于接入页表

200ns 用于内存的字节，合计400ns

b. $0.75\times200+0.25\times400=250ns$

8.10.

① 当页表过大时，可以将未使用的大连续部分折叠一个页表地址为0的单个段表条目

② 分段过长时，可通过分页简化内存分配，从而减少外部碎片造成的内存浪费.

8.11

① 分段基于内存的逻辑划分，因此分段只能与每个用户分段表中的一项共享.

② 分页时页面表为共享的每个页面提供一个公共条目

8.12

a. $219+430=649$

b. $2300+10=2310$

c. 无效引用，超出范围.

d. $1327+400=1727$

e. 无效引用，超出范围

8.13

页表过大时对页表分页可以简化内存分配，并使当前未使用的部分能够交换

8.14

① 当页表本身被分页时，转换页面条目表的位置

② 访问页面表本身

③ 内存加载的操作

8.15

当程序只占大虚拟地址空间的小部分时，首选哈希页表。但当多个页面映射到同一哈希页面表项时，会出现冲突。并且，即使冲突，遍历这个哈希条目对应的列表会导致开销大。

而在分段式分页方案中，这个开销很小，因为每个页表只维护一个相关页面的信息