中山大學 SUN YAT-SEN UNIVERSITY

操作系统笔记：中大 2022 人工智能学院课程

1 并发：死锁与饥饿

1.1 死锁原理

死锁即一组相互竞争系统资源或者通信的进程“永久阻塞”。
相互等待着被对方占有的共享资源，共享资源可以分为可重用资源和可消耗资源。

- 可重用资源
  - 可重用资源指一次仅供一个进程安全使用且不因使用而耗尽的资源。重用即重复使用。进程得到资源单元并使用后会释放供其他进程使用。
  - 如：I/O 通道、内存和外存、设备以及数据结构（文件、数据库和信号量）。
  - 重用资源的竞争可能导致死锁。
- 可消耗资源
  - 可消耗资源指可被创建和销毁的资源。
  - 如：中断、信号、消息和 I/O 缓冲区的信息。

死锁与饥饿

- 死锁是进程在信号量内无穷等待，处于阻塞态。进程量大于 1。
- 饥饿是由于分配策略不公平导致的进程长时间等待，处于阻塞态（等资源）或就绪态（等 CPU）。发生饥饿的进程可以只有一个。

死锁产生的原因

- 系统资源的竞争
- 进程推进顺序非法

死锁产生的必要条件：必须同时成立。 *significant*

- 互斥条件：资源的排他性使用。
- 不可剥夺条件：不可以从其他进程中强行夺走其未释放的资源。
- 请求并保持条件：保持资源时提出新的资源请求，请求不被满足时不释放保持资源。
- 循环等待条件：
  - 存在进程资源的循环等待链，每个进程保持资源被链中的下一个进程所请求。
  - 循环等待的条件比死锁弱，如下图所示。

死锁的处理策略

必须设法破坏产生死锁的四个必要条件，或有能力在死锁产生之后检测并回复。

- 死锁预防：设置限制条件，破坏四个死锁必要条件。
- 避免死锁：在资源动态分配过程中防止系统进入不安全状态。
- 死锁的检测与解除：允许发生死锁，检测并接触死锁。

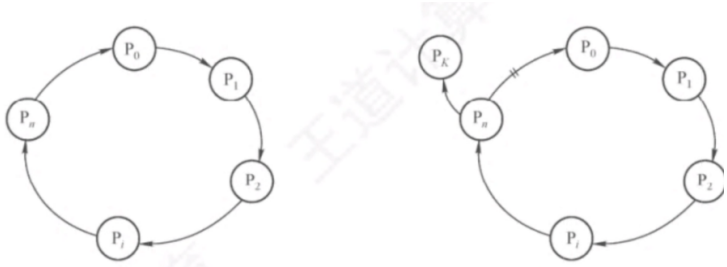

图 1: 循环等待的死锁与非死锁

	资源分配策略	各种可能模式	主要优点	主要缺点
死锁预防	保守，宁可资源闲置	一次请求所有资源，资源剥夺，资源按序分配	适用于突发式处理的进程，不必进行剥夺	效率低，初始化时间延长；剥夺次数过多；不便灵活申请新资源
死锁避免	是预防和检测的折中（在运行时判断是否可能死锁）	寻找可能的安全允许顺序	不必进行剥夺	必须知道将来的资源需求；进程不能被长时间阻塞
死锁检测	宽松，只要允许就分配资源	定期检查死锁是否已经发生	不延长进程初始化时间，允许对死锁进行现场处理	通过剥夺解除死锁，造成损失

图 2: 死锁处理策略比较

1.2 死锁预防

破坏互斥条件：将只能互斥使用的资源改为允许使用的资源。

破坏不可剥夺条件：新的资源不被满足时，进程必须释放已经保持的所有资源。

- 释放已经获得的资源可能导致前一阶段的工作失效。
- 一般适用于易保存和回复的资源，如 CPU 寄存器和内存资源。

破坏请求并保持条件：进程在请求资源的时候不能持有不可剥夺资源。

-

目录与索引

A 附录 1 Assignment

第四次作业

苏睿熹 22330100

- 习题 6.6

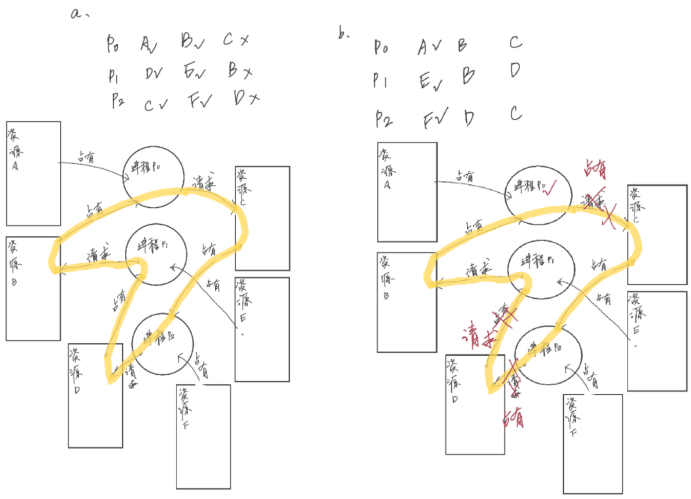

图 3: 习题 6.6 解答

调整后的资源请求顺序：P_0：A⇒B⇒C；P_1：E⇒B⇒D；P_2：F⇒D⇒C

- 习题 6.14

a. 会导致两个全部被阻塞。运行序列：

`foo.semWait(S)`→`bar.semWait(R)` →`foo.semWait(R)`→`bar.semWait(S)`

b. 可能会导致其中一个被无限期延后。运行序列：

```
foo.semWait(S); foo.semWait(R); x++;
foo.semSignal(S); bar.semWait(R); foo.semSignal(R);
foo.semWait(S); foo.semWait(R); x++;
……
```

- 习题 6.15
  - 目前已经分配的资源数量有：$1+1+3+2=7$，假设目前可用资源为 $x \geq 1$。
  - 因为 $x \geq 1$，可执行 P_1 并释放。可用资源为 $x+1 \geq 2$。
  - 因为 $x+1 \geq 2$，可执行 P_0 并释放，可用资源为 $x+2$。
  - 目前 P_2 和 P_3 分别需要 6 个和 5 个资源，优先执行 P_3。
  - 要使得 P_3 可执行，可用资源满足 $x+2 \geq 5$。

  - 释放 P_3 得可用资源为 $x+4 \geq 7$，可执行 P_2。
  - 所以，要保证当前状态的安全性，最少需要 $\min\{x\}+7=10$ 个单位的资源。
- 习题 7.5

a. 优点：最差适配算法倾向于选择最大的空闲存储块来放置进程。这样可以避免在较小的空闲块中留下大量的小碎片。通过使用较大的空闲块，减少了内部碎片的数量，从而提高了内存的整体利用率。缺点：由于需要找到最大的空闲块，因此每次分配时都需要遍历整个可用内存列表，导致查找时间较长。如果频繁地为小进程分配大的空闲块，可能会导致大量未使用的内存被浪费。

b. 最差适配的平均查找长度取决于内存中的空闲块数量和大小分布情况。一般来说，由于每次分配都要遍历所有空闲块以找到最大的一个，所以平均查找长度会随着空闲块数量的增加而增加。

- 习题 7.12
  - 逻辑地址有多少位？

$$\text{逻辑地址} = \log_2(2^{32}) + \log_2(2^{10}) = 32 + 10 = 42 \text{ bits}$$

  - 一个页框有多少字节？

$$\text{页框大小} = 2^{10} \text{ bytes} = 1024 \text{ bytes}$$

  - 物理地址中有多少位是页框号？

$$\text{物理地址} = \log_2(2^{32}) = 32 \text{ bits}$$

$$\text{页框号} = \frac{\text{物理地址}}{\text{页框大小}} = \frac{32}{10} = 3.2 \approx 3 \text{ bits}$$

  - 页表中有多少表项？

$$\text{页表项数} = 2^{16}$$

  - 假设每个页表项中含一位有效位，每个页表项有多少位？

$$\text{页表项大小} = \log_2(2^{16}) + 1 = 16 + 1 = 17 \text{ bits}$$