

8.3

- **First-fit:**

212 KB 放入 500 KB partition
417 KB 放入 600 KB partition
112 KB 放入 288 KB (500 - 212) partition
426 KB 等待

- **Best-fit:**

212 KB 放入 300 KB partition
417 KB 放入 500 KB partition
112 KB 放入 200 KB partition
426 KB 放入 600 KB partition

- **Worst-fit:**

212 KB 放入 600 KB partition
417 KB 放入 500 KB partition
112 KB 放入 388 KB partition
426 KB 等待

Best-fit 利用率最高

8.5

a. external fragmentation

- **连续内存分配:** 由于地址空间是连续分配的, 当旧进程结束、新进程启动时会产生空洞, 因此会受到**外部碎片**的影响。
- **纯分段:** 由于一个进程的段必须连续分配在物理内存中, 因此当旧进程的段被新进程的段替换时也会出现碎片, 因而同样会受到**外部碎片**的影响。
- **纯分页:** 不会受到外部碎片的影响, 因为页是以固定大小分配的, 物理内存可以不连续。

b. internal fragmentation

- **连续内存分配:** 由于内存分配是按照进程所需的大小进行的, 不会有显著的**内部碎片**。
- **纯分段:** 与连续内存分配类似, 只会为每个段分配需要的大小, 因此**内部碎片**较小。
- **纯分页:** 由于内存以页为单位分配, 如果某页未完全使用, 则会产生**内部碎片**, 导致空间浪费。

c. ability to share code across processes

- **连续内存分配**: 不支持代码共享, 因为进程的虚拟地址空间是连续的, 无法划分成非连续的小块。
- **纯分段**: 支持代码共享。例如, 两个不同的进程可以共享同一个代码段, 但它们会有各自独立的数据段。
- **纯分页**: 支持代码共享。例如, 两个进程可以共享同一页内的代码内容。

8.9

a.

$$200 + 200 = 400ns$$

b.

$$200 \times 0.75 + 400 \times 0.25 = 250ns$$

8.12

a.

$$219 + 430 = 649$$

b.

$$2300 + 10 = 2310$$

c.

$$500 > 100, \text{ 不合法}$$

d.

$$1327 + 400 = 1727$$

e.

$$112 > 96, \text{ 不合法}$$