操作系统

Operating system

孔维强

大连理工大学

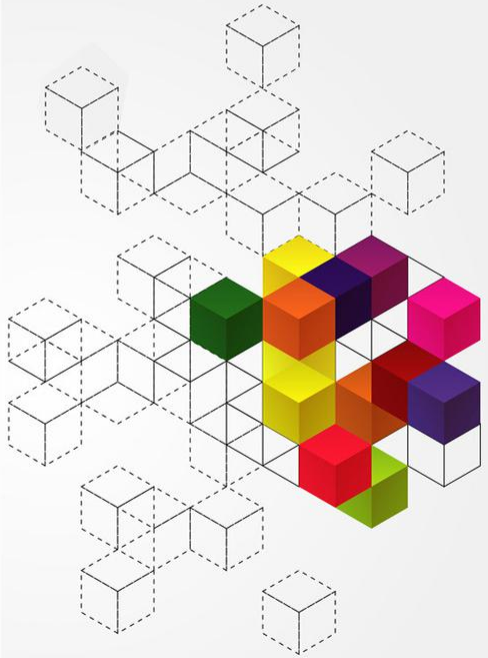
内容纲要

6.8 读者写者问题

一、 读者写者问题简介

二、 进程协作关系分析

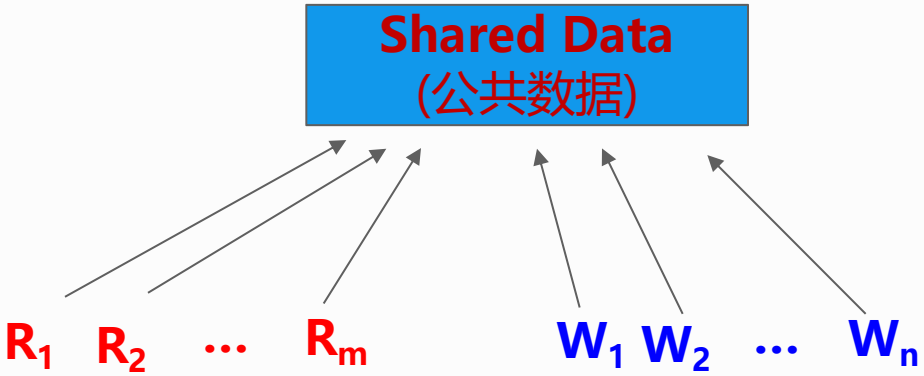
三、 读者写者问题的同步解法

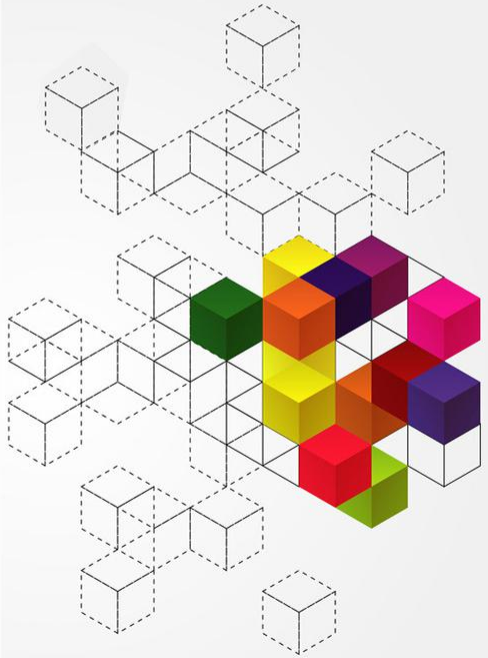


一、读者写者问题简介

一、读者写者问题简介

- Reader-Writers Problems
  - 公共数据
  - Reader
  - Writer





二、进程协作关系分析

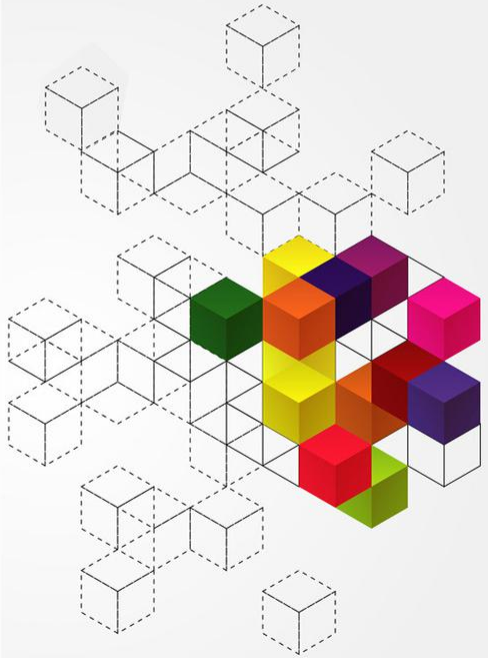
步骤1：读者-写者进程表示

```
R:
    while (true) {
        read();
    }
```

```
W:
while (true) {
    write();
}
```

步骤2：进程协作关系分析

- 读操作可以同时进行（R-R，共享读）
- 读操作和写操作不可以同时进行（R-W，互斥）
- 写操作和写操作不可以同时进行（W-W，互斥）



三、同步解法

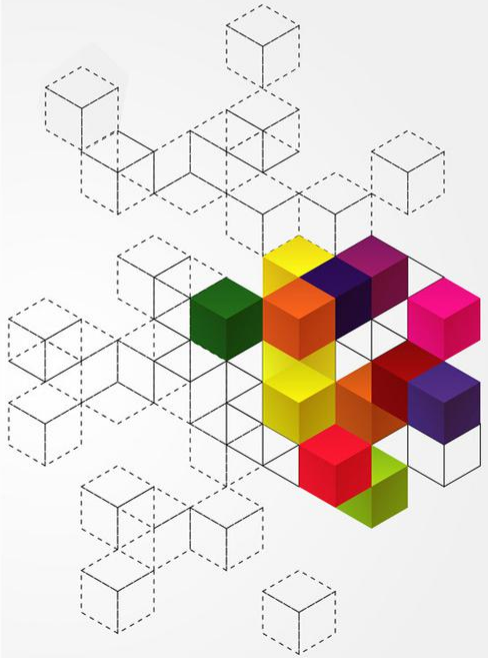
- Try1: 使用信号量mutex对读写操作进行互斥保护

```
R:
    while (true)  {
        P(mutex);
        read();
        V(mutex);
    }
```

```
W:
while (true)  {
    P(mutex);
    write();
    V(mutex);
}
```

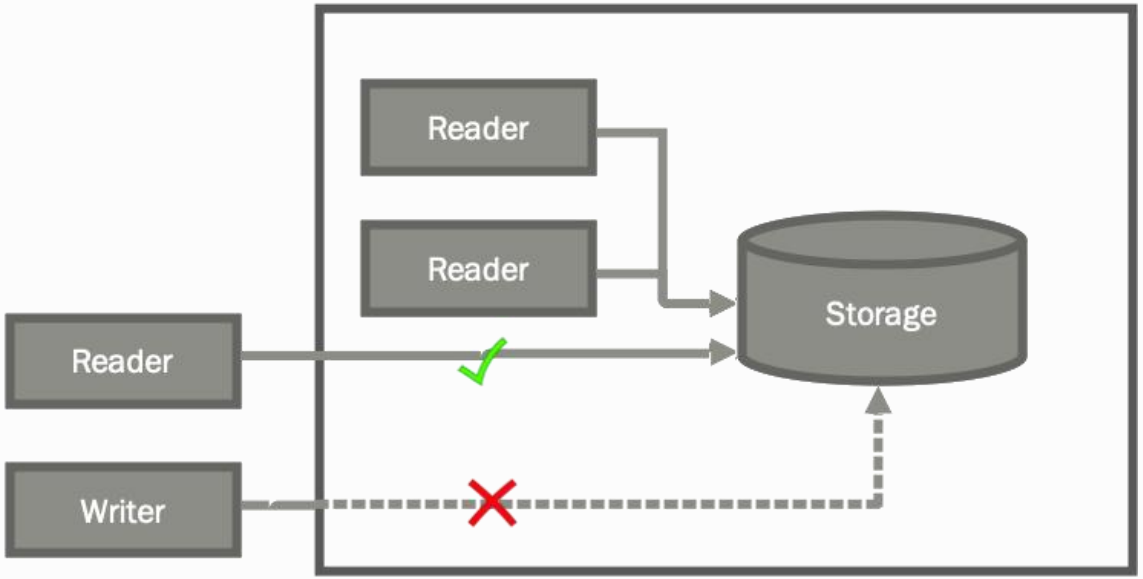
- mutex初值为1

- 是否正确解决问题?

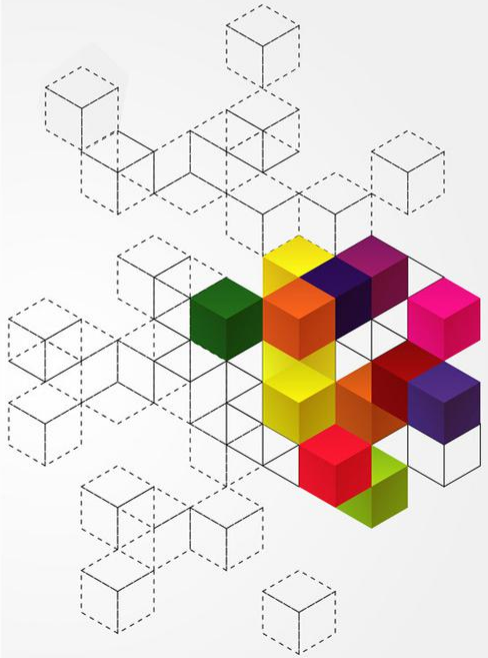


三、同步解法

- Try1: 使用信号量mutex对读写操作进行互斥保护
  对读者之间施加了互斥，过于严苛



- 对try1进行改进，重点考虑保证R-R共享



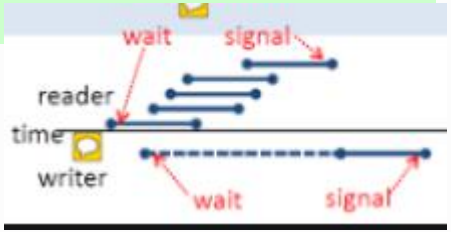
三、同步解法

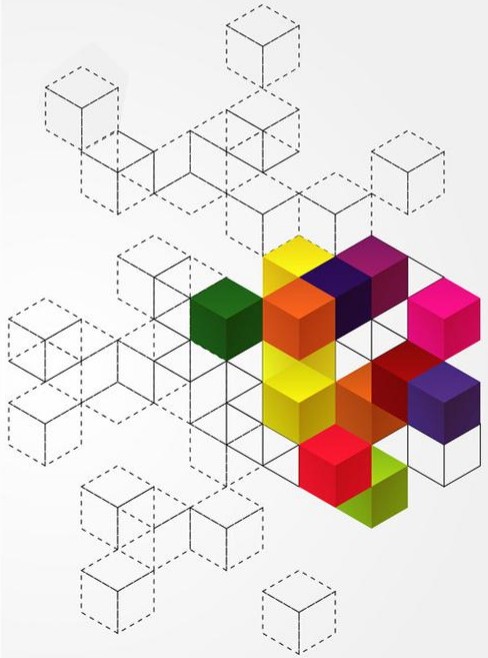
- Try2: 通过引入引用计数保证读者共享

```
Reader:
  while (true)  {
     P(r_mutex);
     r_cnt++;
     if(r_cnt==1)
        P(mutex);
     V(r_mutex);
     read();
     P(r_mutex);
     r_cnt- -;
     if(r_cnt==0)
        V(mutex);
     V(r_mutex);
  }
```

```
Writer:
while (true)  {
    P(mutex);
    write();
    V(mutex);
};
```

问题：Writer Starvation





三、同步解法

- Try3: 引入额外的rw_mutex用于R-W竞争

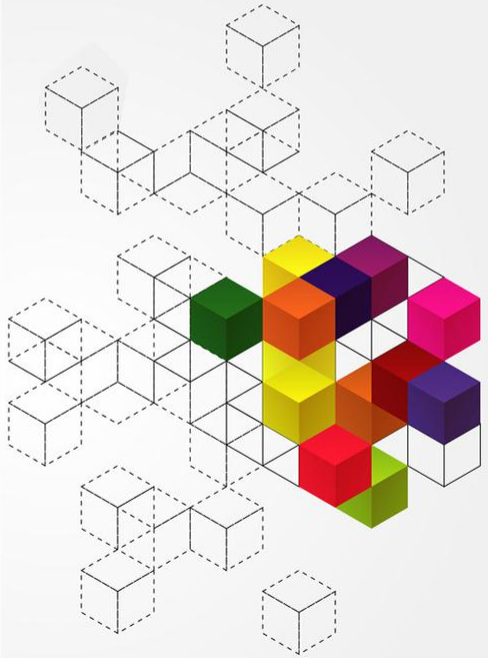
Final Solution

R:

```
while (true)  {
    P(rw_mutex);
    P(r_mutex);
    r_cnt++;
    if(r_cnt==1)
     P(mutex);
    V(r_mutex);
    V(rw_mutex);
    read();
    P(r_mutex);
    r_cnt- -;
    if(r_cnt==0) V(mutex);
    V(r_mutex);
}
```

W:

```
while (true)  {
    P(rw_mutex);
    P(mutex);
    write();
    V(mutex);
    V(rw_mutex);
};
```



本讲小结

- 读者写者问题同步分析
- 基于信号量的读者写者问题同步解法