# Операционные системы

# Примеры взаимоисключений

Олег Французов

2017

# Производители и потребители

- K процессов, производящих данные
- L процессов, обрабатывающих данные
- Буфер на N порций данных

# Производители и потребители

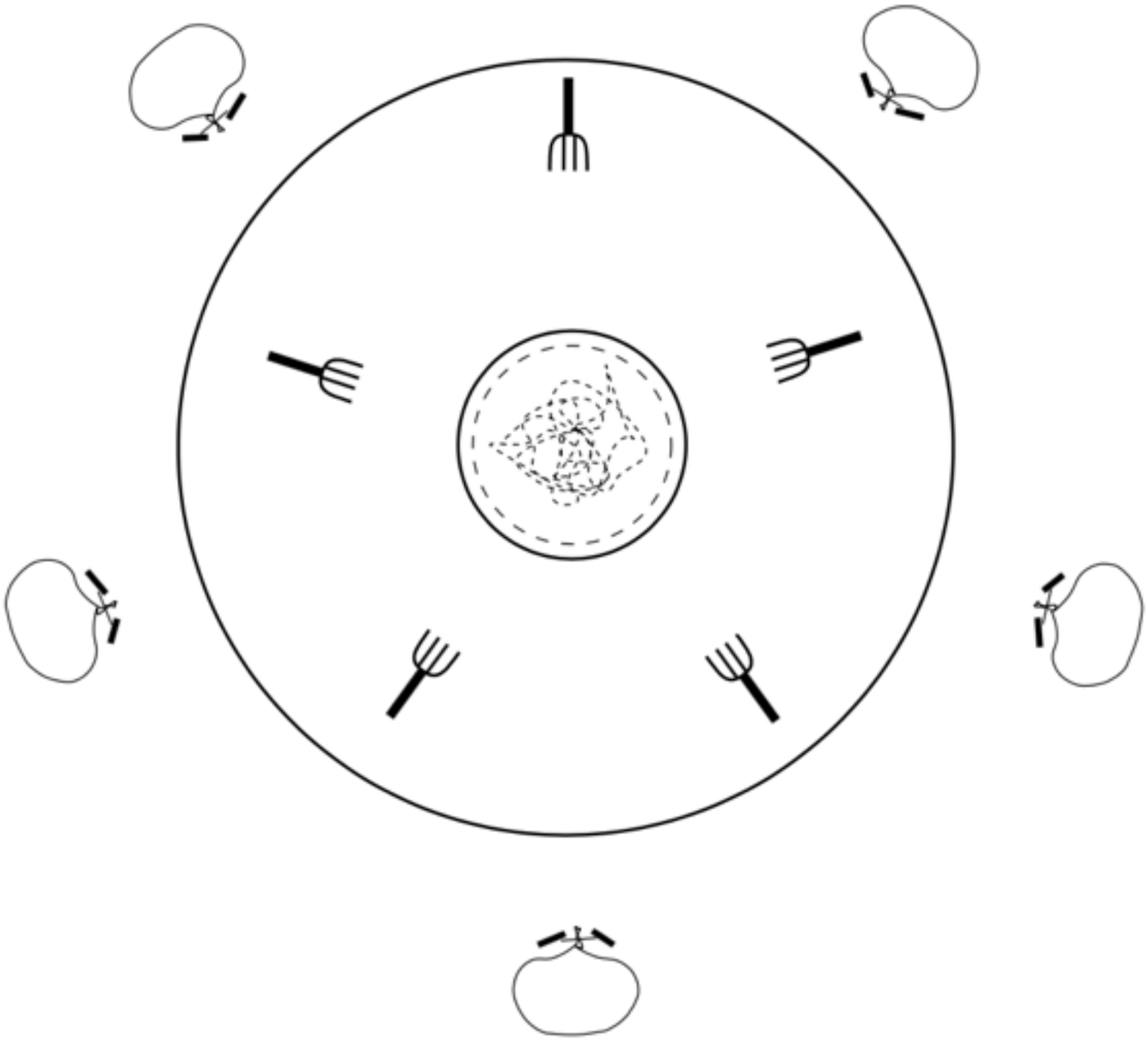
- Взаимное исключение доступов к буферу
- Буфер пуст, блокировать потребителей
- Буфер полон, блокировать производителей

# Производители и потребители

```
void producer() {
  /* ... подготовить
        данные ...
   */
  down(empty);
  lock(m);
  put_data();
  unlock(m);
  up(full);
}
```

```
void consumer() {
  down(full);
  lock(m);
  get_data();
  unlock(m);
  up(empty);
  /* ... обработать
        данные ...
   */
}
```

# Обедающие философы



# Тупик (deadlock)

- Все участвующие процессы заблокировались в ожидании друг друга

# Более простой пример

```
lock(m1);
lock(m2);
```

```
lock(m2);
lock(m1);
```

# Пример без мьютексов

```
char buf[100];
int rc;
int fd[2];
pipe(fd);
if(fork()==0) {
    dup2(fd[1], 1);
    close(fd[1]);
    close(fd[0]);
    execlp("ls", "ls", NULL);
    perror("ls");
    exit(1);
}
close(fd[1]);
wait(NULL);
while((rc = read(fd[0], buf, sizeof(buf)))>0) {
    /* ... */
}
```

# Обедающие философы

```
int left(int n) { return (n - 1 + 5) % 5; }
int right(int n) { return (n + 1) % 5; }
```

# Обедающие философы

```
void philosopher(int n) {
    for(;;) {
        think();
        lock(forks[n]);              /* ! */
        lock(forks[right(n)]);
        eat();
        unlock(forks[n]);
        unlock(forks[right(n)]);
    }
}
```

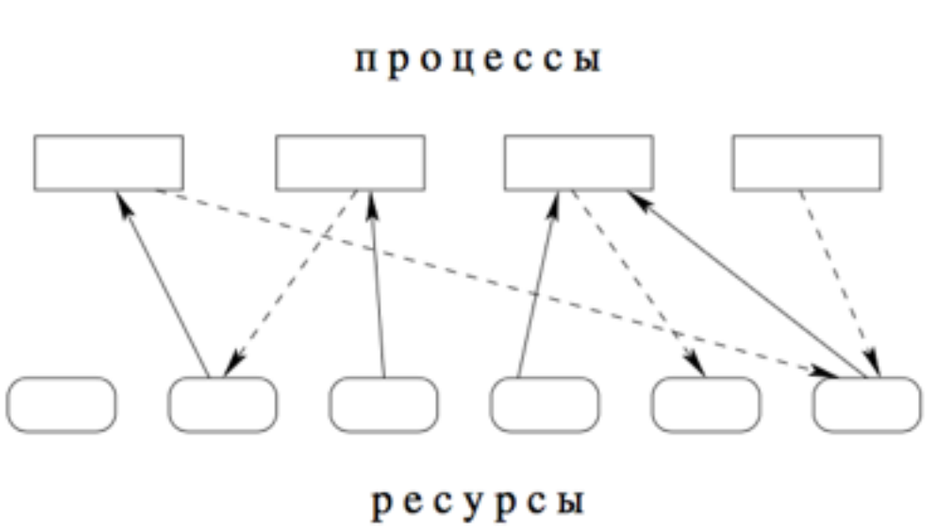
# Обедающие философы

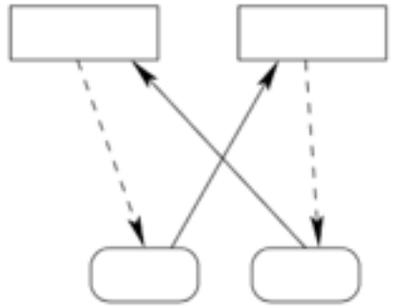
```
void philosopher(int n) {
    for(;;) {
        think();
        down(sem);
        lock(forks[n]); lock(forks[right(n)]);
        eat();
        unlock(forks[n]); unlock(forks[right(n)]);
        up(sem);
    }
}
```

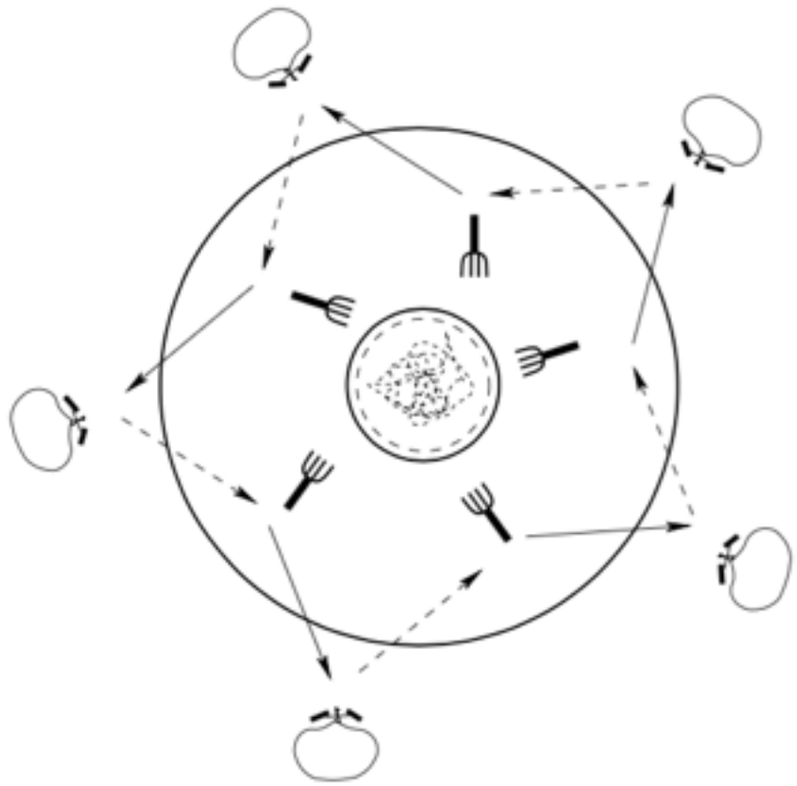
# Обедающие философы

- c. 168

# Граф ожидания





- Тупик ~ цикл в графе ожидания

# Читатели и писатели

- Либо есть N>0 читателей
- Либо ровно один писатель

# Читатели и писатели

```
void writer(...) {
    lock(db_mutex);
    /* ... пишем данные в общую память ... */
    unlock(db_mutex);
}
```

```
void reader(...) {
    lock(rc_mutex);
    rc++;
    if(rc == 1) lock(db_mutex); /* первый! */
    unlock(rc_mutex);
    /* ... читаем данные из общей памяти ... */
    lock(rc_mutex);
    rc--;
    if(rc == 0) unlock(db_mutex); /* уходя, гасите свет */
    unlock(rc_mutex);
}
```

# Q & A