

Задача 6. Предложить программу, реализующую процессы медведя и n пчел.

Исследование провёл студент группы 22207 Гордеев Никита

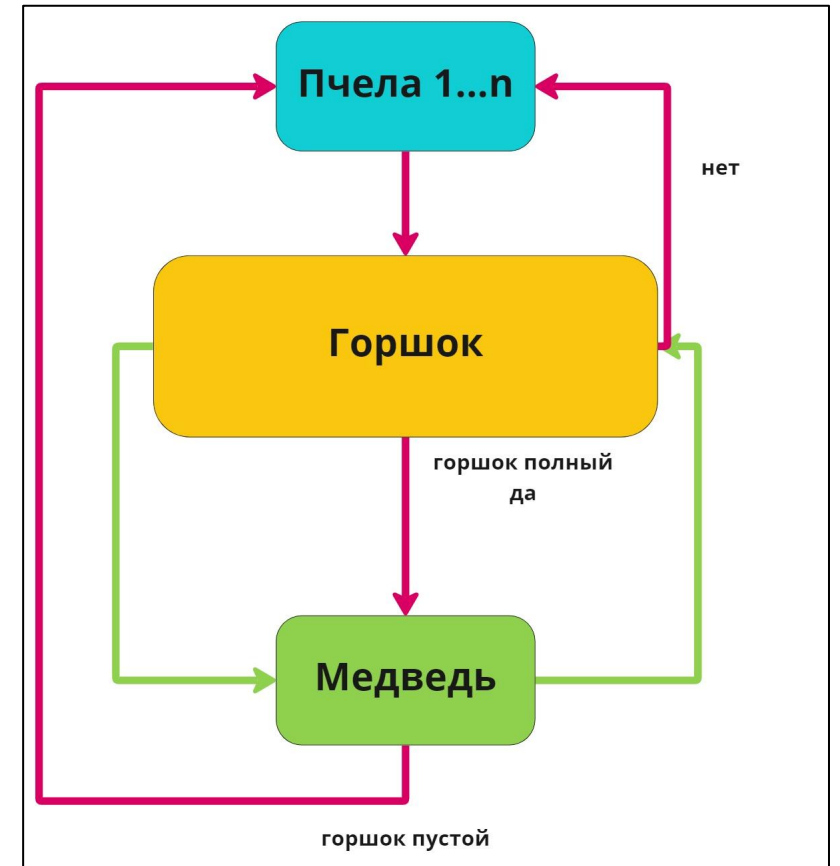
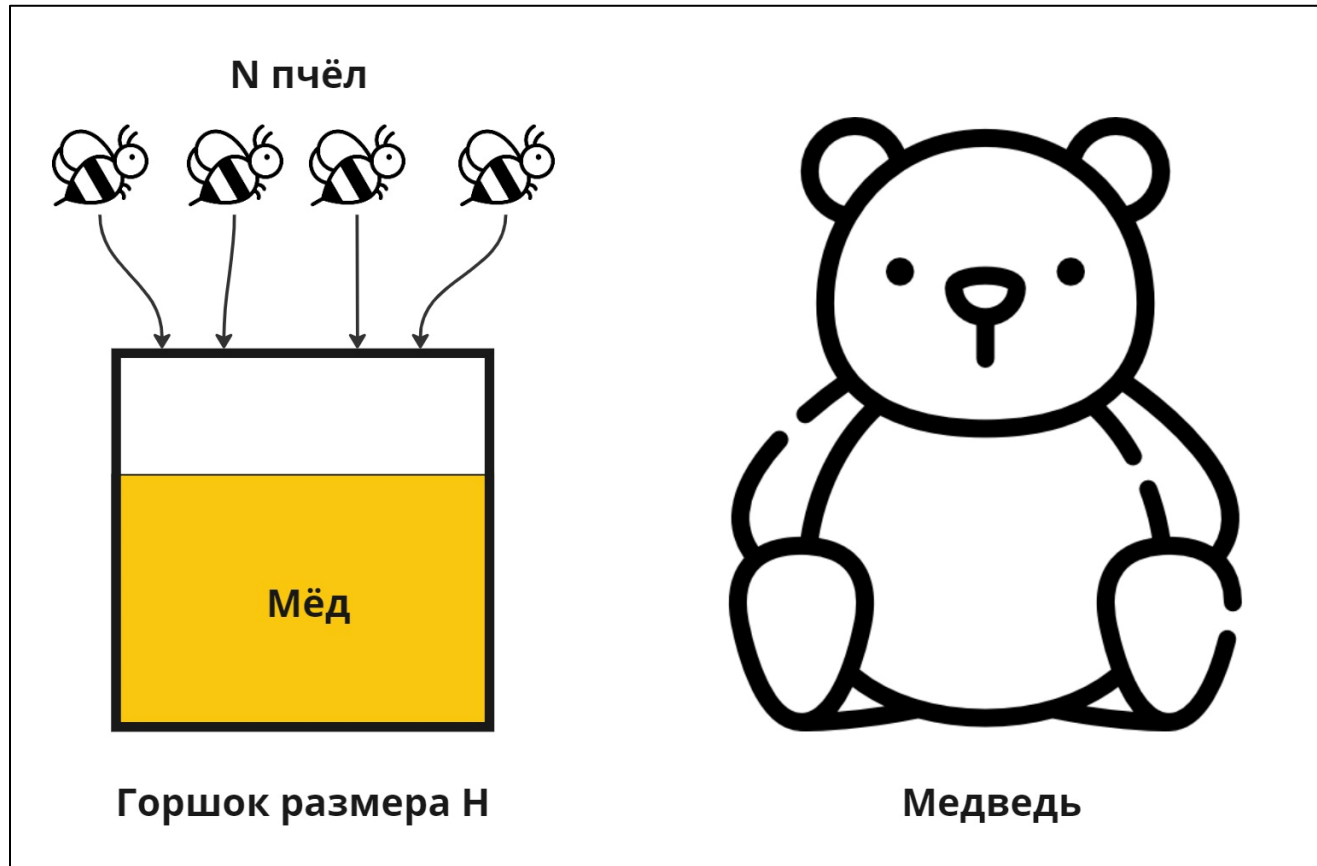
Дата выполнения работы 18.12.2022 (Вариант 2)

Постановка задач:

Медведь и пчёлы. Есть n пчёл и медведь. Они пользуются одним горшком мёда, вмещающим N порций мёда. Сначала горшок пустой. Пока горшок не наполнится, медведь спит, потом съедает весь мёд и засыпает. Каждая пчела многократно собирает одну порцию мёда и кладёт её в горшок. Пчела, которая приносит последнюю порцию мёда и заполняет горшок, будит медведя.

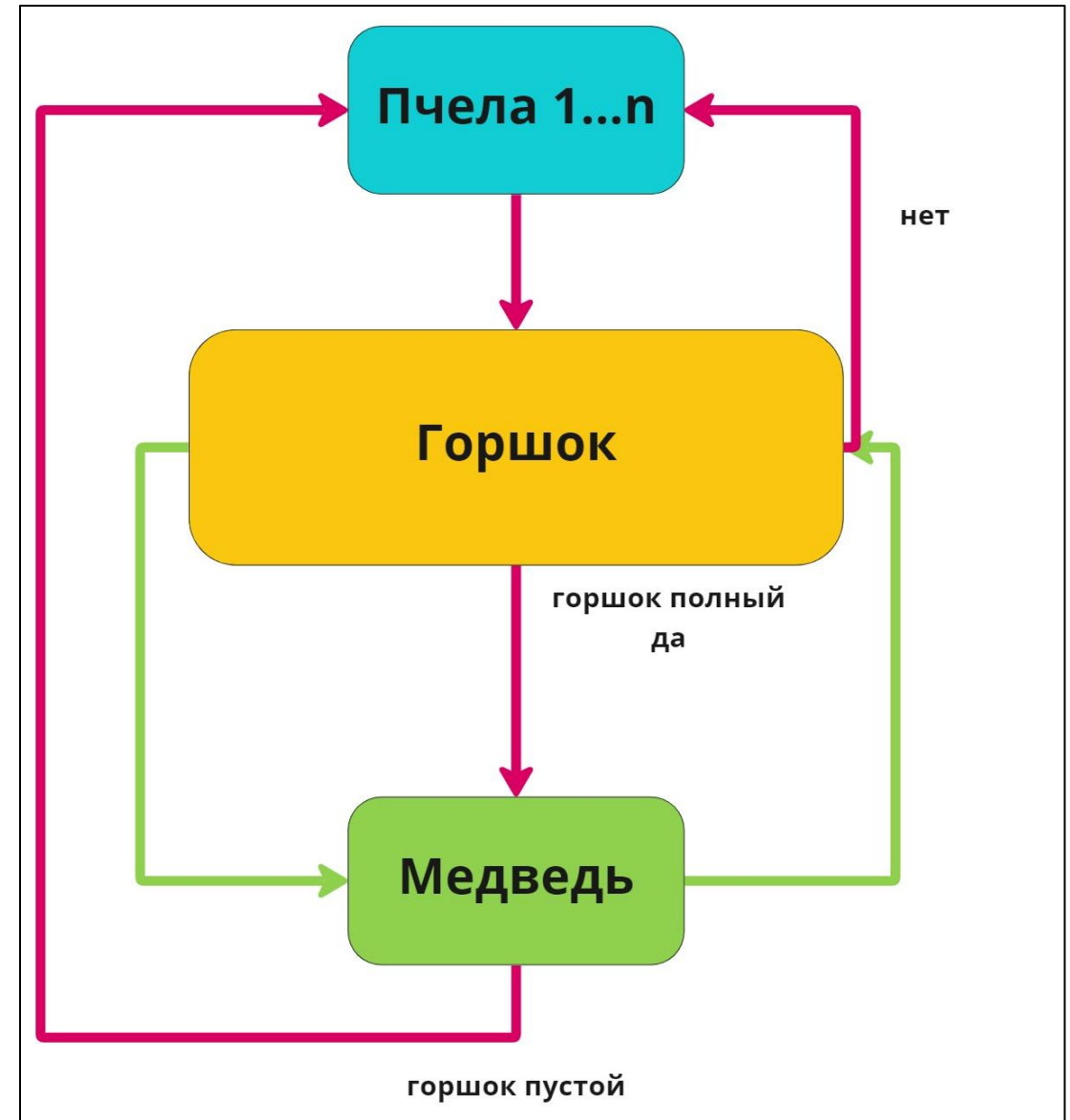
1. Представьте медведя и пчёл процессами
2. Разработайте код, моделирующий их действия
3. Для синхронизации используйте семафоры

Визуализация условия:



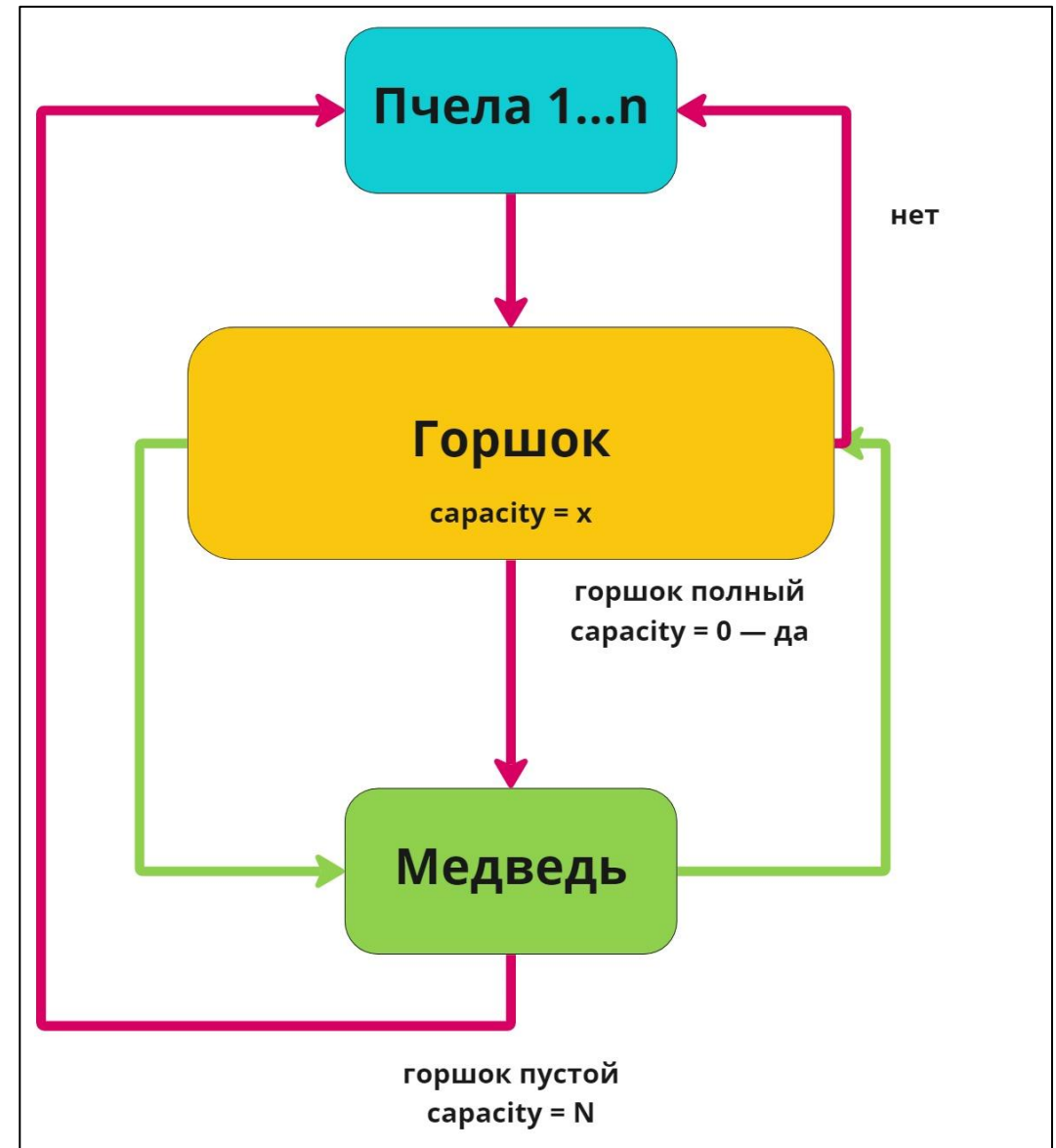
Критическая секция

- КС возникает в момент, когда горшок полный.
- Остальные пчелы не должны приносить мед
- Будить медведя должна одна пчела.



Идеи:

- Представим медведя как один процесс и пчёл как n одинаковых процессов
- Никакие два процесса (пчела/пчела или пчела/медведь) не могут одновременно использовать горшок
- Объявим семафор `capacity` который будет означать сколько меда еще надо положить в горшок
- Пчелы с помощью операции `P(capacity)` будут класть мед в горшок, если это возможно, иначе ждать
- Если после того, как пчела положила мед и `capacity = 0`, она будит медведя
- Медведь устанавливает `capacity = n`



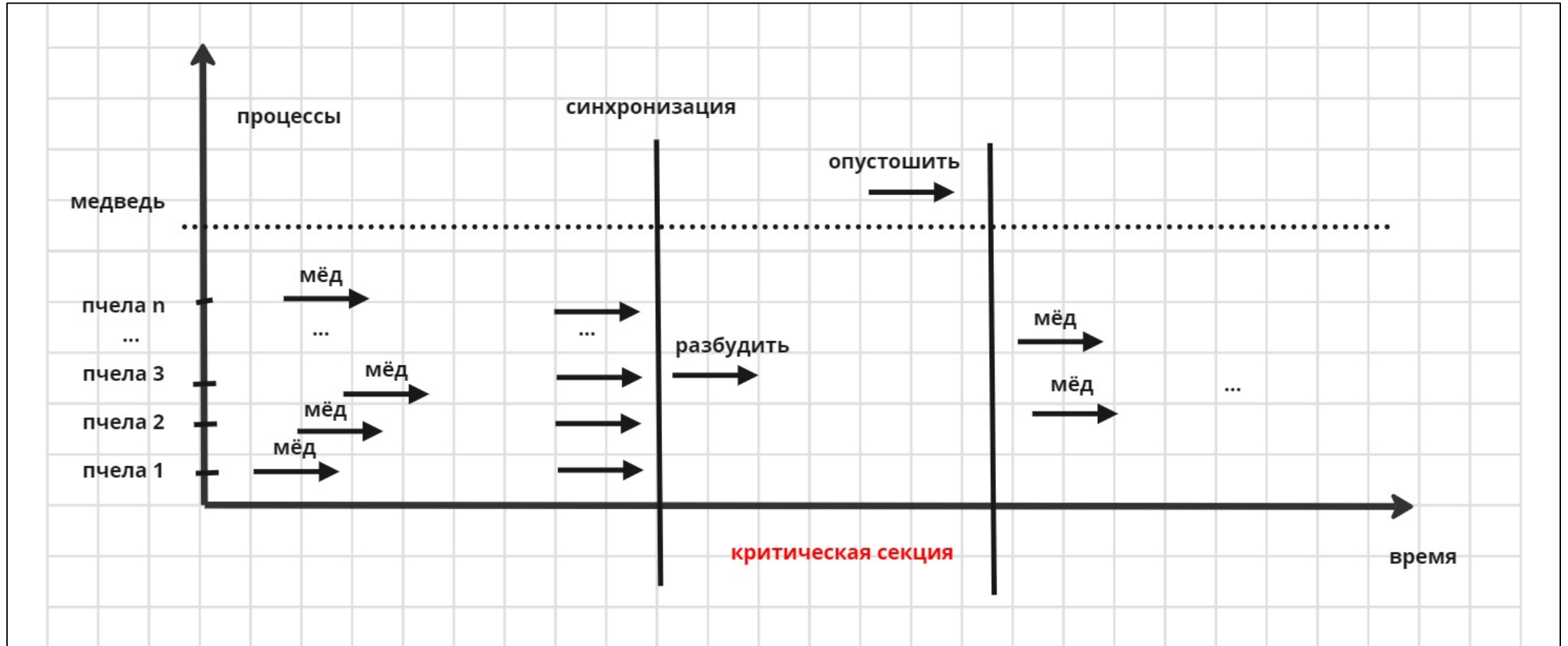
Программа реализующую процессы медведя и n пчел

```
int have_honey = 0; # Сколько мёда в горшке
int lock_put = 0; # Можно класть мёд
int lock_eat = 1; # Нельзя есть мёд

process Bee[i = 1 to n] {
  ____while(1) {
    _____# Блокируем другие процессы-пчёлы
    _____<await (lock_put == 0) lock_put = 1;>
    _____have_honey++;
    _____if (have_honey == n) # Если горшок полон
    _____lock_eat = 0; # Можно есть мёд
    _____else lock_put = 0; # Иначе снимаем
    _____блокировку других процессов-пчёл
    _____}
  ____}
}
```

```
process Bear {
  while(1) {
    _____# Процесс заблокирован, пока замок не снят
    _____<await (lock_eat == 0)
    have_honey = 0; # Съесть мёд
    lock_eat = 1; # Замок на поедание мёда
    _____# Разблокируем процессы-пчёлы
    lock_put = 0;>
  }
}
```

Схема работы



Анализ

- **Семафоры обеспечивают синхронизацию процессов: когда горшок уже заполнен, выполнение процессов-пчел приостановлено.**
- **Только одна пчела разбудит медведя, т.к. остальные сразу заблокируются.**

Материалы:

- Эндрюс Г.Р. Основы многопоточного, параллельного и распределенного программирования / Г.Р.Эндрюс. - Москва : Вильямс, 2003. - 512 с
- Bee free icon // flaticon URL: https://www.flaticon.com/free-icon/bee_4208241?term=bee&page=1&position=41&page=1&position=41&related_id=4208241&origin=search (дата обращения: 22.10.2022).
- Teddy Bear free icon // flaticon URL: https://www.flaticon.com/free-icon/teddy-bear_1047561?term=bear&page=1&position=20&page=1&position=20&related_id=1047561&origin=tag (дата обращения: 22.10.2022).

Изменения

- **Версия 2**
 - Добавил код программы
 - Добавил схемы работы программы
 - Расширил анализ