

### **Билет №1 Задача 1**

Задача об обедающих философах - прокомментируйте:

Философы сидят за круглым столом. У каждого спагетти, но чтобы их есть нужны 2 вилки. Философ может либо размышлять, либо голодать, либо есть.

```
procedure typical_phylosophers;
begin
  while true do
    begin
      think_any_time;      // размышляем
      repeat
        take left_fork;    //пытаемся захватить левую вилку
        if right_fork absent then plase left_fork // если правая вилка занята – кладем левую
        else take right_fork; // в прот. случае берем и правую
      until bouth_fork_hand; // повторяем, пока не захватим обе вилки
      eat_some_time; // спагетти кушаем
      place left_fork; // кладем левую вилку
      place right_fork; // кладем правую вилку
    end;
  end;
```

Минус алгоритма – все философы одновременно берут левую вилку. И не найдя правую, кладут левую, потом снова все берут левую и т.д. – зависание. Если вставить в алгоритм задержку на случайный интервал времени после того как положили вилки, то все будет хорошо. Этот способ используется, например, при коллизиях в сетях Ethernet. Однако подобный выход не совсем надежен (напр., на атомных АС играть со случайными числами опасно – рванет уже сто раз).

Если захват вилок, подание и освобождение вилок защитить бинарным семафором – все будет работать, но одновременно есть сможет лишь 1 философ. Выход – каждому философу по семафору и по переменной, отслеживающей состояние философа (думает, голоден, ест). И еще 1 бин. семафор на вход в крит. секцию – выявление голодного философа и попытку получить вилку.

### **Билет №2 Задача 1**

Задача об обедающих философах - прокомментируйте:

Философы сидят за круглым столом. У каждого спагетти, но чтобы их есть нужны 2 вилки. Философ может либо размышлять, либо голодать, либо есть.

```
procedure typical_phylosophers;
begin
  while true do
    begin
      think_any_time; // размышление
      take bouth_fork_at_once; // пытается захватить ОБЕ вилки ОДНОВРЕМЕННО
      eat_some_times; // съест спагетти
      place bouth_fork_at_once; // кладет обе вилки
    end;
  end;
```

Недостаток алгоритма - часть философов покушает, часть сдохнет от голода

### **Билет №6 Задача 3**

Задача об обедающих философах - прокомментируйте:

Философы сидят за круглым столом. У каждого спагетти, но чтобы их есть нужны 2 вилки. Филосов может либо размышлять, либо голодать, либо есть.

```
procedure typical_phylosophers;
begin
  while true do
    begin
      think_any_time; // думает
      take left_fork; // пытается взять левую вилку
      take right_fork; // пытается взять правую
      eat_some_times; // спагетти едим
      place bouth_fork_at_once; // кладем обе вилки
    end;
  end;
```

Минус алгоритма – все философы одновременно берут левую вилку, но правую вилку взять уже не смогут – она у соседа. Тупик.

### **Билет №7 Задача 3**

Задача об обедающих философах - прокомментируйте:

Философы сидят за круглым столом. У каждого спагетти, но чтобы их есть нужны 2 вилки. Филосов может либо размышлять, либо голодать, либо есть.

```
procedure typical_phylosophers;
begin
  while true do
    begin
      think_any_time; // размышляем
      repeat
        take left_fork; //пытаемся захватить левую вилку
        if right_fork absent then plase left_fork // если правая вилка занята – кладем левую
        else take right_fork; // в прот. случае берем и правую
      until bouth_fork_hand; // повторяем, пока не хахватим обе вилки
      eat_some_time; // спагетти ням-ням
      place left_fork; // кладем левую вилку
      place right_fork; // кладем правую вилку
    end;
  end;
```

Минус алгоритма – все философы одновременно берут левую вилку. И не найдя правую, кладут левую, потом снова все берут левую и т.д. – зависание. Если вставить в алгоритм задержку на случайный интервал времени после того как положили вилки, то все будет хорошо. Этот способ используется, например, при коллизиях в сетях Ethernet. Однако подобный выход не совсем надежен (напр., на атомных АС игратья со случайными числами опасно – рванет уже сто раз).

Если захват вилок, подание и освобождение вилок защитить бинарным семафором – все будет работать, но одновременно есть сможет лишь 1 философ. Выход – каждому философу по семафору и по переменной, отслеживающей состояние философа (думает, голоден, ест). И еще 1 бин. семафор на вход в крит. секцию – выявление голодного философа и попытку получить вилку.

### **Билет №10 Задача 3**

Задача об обедающих философах - прокомментируйте:

Философы сидят за круглым столом. У каждого спагетти, но чтобы их есть нужны 2 вилки. Филосов может либо размышлять, либо голодать, либо есть.

```
procedure typical_phylosophers;  
begin  
  while true do  
    begin  
      think_any_time; // думает  
      take_left_fork; // пытается взять левую вилку  
      take_right_fork; // пытается взять правую  
      eat_some_times; // спагетти ням-ням  
      place_bouth_fork_at_once; // кладем обе вилки  
    end;  
  end;  
end;
```

Минус алгоритма – все философы одновременно берут левую вилку, но правую вилку взять уже не смогут – она у соседа. Тупик

### **Билет №11 Вопрос 3**

Задача об обедающих философах - прокомментируйте:

Философы сидят за круглым столом. У каждого спагетти, но чтобы их есть нужны 2 вилки. Филосов может либо размышлять, либо голодать, либо есть.

```
procedure typical_phylosophers;  
begin  
  while true do  
    begin  
      think_any_time; // думает  
      take_left_fork; // пытается взять левую вилку  
      take_right_fork; // пытается взять правую  
      eat_some_times; // спагетти ням-ням  
      place_bouth_fork_at_once; // кладем обе вилки  
    end;  
  end;  
end;
```

Минус алгоритма – все философы одновременно берут левую вилку, но правую вилку взять уже не смогут – она у соседа. Тупик.