

27. **Задача про экзамен.** Преподаватель проводит экзамен у группы студентов. Каждый студент получает свой билет, сообщает его номер и готовит письменный ответ. Подготовив ответ, он передает его преподавателю. Преподаватель просматривает ответ и сообщает студенту оценку. Студент, дождавшись результата, уходит с экзамена. *Требуется создать приложение, моделирующее действия преподавателя и студентов, каждый из которых представлен отдельным процессом.*

#### 4 балла (именованные POSIX семафоры)

##### Файлы и каталоги:

program\_4\_poins-ans.c

4\_points\_test.zip

##### Ввод и вывод данных:

```
mastavtsev@mastavtsev-VirtualBox:~/IHT_2/4_points$ gcc program_4_poins-ans.c
mastavtsev@mastavtsev-VirtualBox:~/IHT_2/4_points$ ./a.out 2
Professor received a variant from Student 2:
  Student 2: my variant is 16
Professor received an answer from Student 2:
  Student 2: my answer is ANSWER 9
Student 2 received a mark from professor:
  Your mark is 4!
Professor received a variant from Student 1:
  Student 1: my variant is 15
Professor received an answer from Student 1:
  Student 1: my answer is ANSWER 8
Student 1 received a mark from professor:
  Your mark is 7!
```

Параметрами для исполняемого файла являются:

- 1) N – количество детей у дочернего процесса. В рамках решаемой задачи (и не только ☺) дети – это студенты, пришедшие сдавать экзамен процессу родителю – профессору.

## Процесс сдачи экзамены (выходные данные):

Данный раздел актуален для всех программ, поэтому пояснения будут только тут.

Процесс сдачи экзамен происходит следующим образом:

1. Студент тянет билет -> сообщает номер варианта профессору (через канал/способ передачи информации, обозначенный в задании на конкретный балл).
2. Профессор получает сообщение от студента, выводит его и продолжает «принимать экзамен» - ожидает сообщения от других студентов.
3. После сообщения варианта студент «решает свой билет» и сообщает ответ преподавателю, так же необходимым в задаче способом (ответом является просто фраза ANSWER “random num”). Далее он ждёт, когда профессор сообщит ему оценку.
4. Профессор думает над оценкой студента (sleep(1)) и сообщает ему.
5. Студент, получив оценку от профессора, завершает работу.

Все необходимы синхронизации процессов происходят за счёт семафоров, обозначенных в задании.

Передача сообщений от студента к профессору происходит на основе структуры данных Message:

student\_id - уникальный номер студента, используется для различия студентов

message\_type - тип сообщения:

message\_type == 0 - сообщение о номере варианта

message\_type == 1 - ответ на вариант

message - само сообщение

От студента к профессору идёт обычная строка, кроме программ на 9 и 10 баллов, там так же передача основывается на структуре Message.

В конце работы родительского процесса он дожидается работы дочерних процессов, и после производит освобождение ресурсов. Освобождение

ресурсов происходит также при прерывании программы по нажатию Ctrc+C.

## **5 баллов (неименованные POSIX семафоры)**

### **Файлы и каталоги:**

program\_5\_poins-ans.c

### **Ввод и вывод данных:**

```
mastavtsev@mastavtsev-VirtualBox:~/IHT_2/5 points$ gcc program_5_poins-ans.c
mastavtsev@mastavtsev-VirtualBox:~/IHT_2/5 points$ ./a.out 2
Professor received a variant from Student 2:
  Student 2: my variant is 20
Professor received an answer from Student 2:
  Student 2: my answer is ANSWER 20
Student 2 received a mark from professor:
  Your mark is 8!
Professor received a variant from Student 1:
  Student 1: my variant is 19
Professor received an answer from Student 1:
  Student 1: my answer is ANSWER 19
Student 1 received a mark from professor:
  Your mark is 2!
```

Параметрами для исполняемого файла являются:

- 1) N – количество детей у дочернего процесса.

### **Общая схема решаемой задачи:**

Отличием в реализации от задачи на 4 балла является то, что неименованные семафоры так же хранятся в разделяемой памяти. Остальное взаимодействие совпадает.

## 6 баллов (семафоры System V)

### Файлы и каталоги:

program\_6\_poins-ans.c

### Ввод и вывод данных:

```
mastavtsev@mastavtsev-VirtualBox:~/IHT_2/6_points$ gcc program_6_poins-ans.c
```

```
Professor received a variant from Student 2:  
    Student 2: my variant is 8  
Professor received an answer from Student 2:  
    Student 2: my answer is ANSWER 13  
Student 2 received a mark from professor:  
    Your mark is 1!  
Professor received a variant from Student 1:  
    Student 1: my variant is 7  
Professor received an answer from Student 1:  
    Student 1: my answer is ANSWER 12  
Student 1 received a mark from professor:  
    Your mark is 5!
```

Параметрами для исполняемого файла являются:

- 1) N – количество детей у дочернего процесса.

### Общая схема решаемой задачи:

Данная программа уже реализует использование семафоров System V, также как и разделяемую память в формате System V.

## 7 баллов (именованные семафоры POSIX)

### Файлы и каталоги:

professor.c – программа профессора

student.c – программа студента

p.out – готовый исполняемый файл для профессора

s.out – готовый исполняемый файл для студента

2\_students.sh – скрипт для запуска двух студентов

### Ввод и вывод данных:

```
mastavtsev@mastavtsev-VirtualBox:~/IHT_2/7_points$ ./p.out 2
Professor received a variant from Student 1:
    Student 1: my variant is 18
Professor received an answer from Student 1:
    Student 1: my answer is ANSWER 18
Professor received a variant from Student 1:
    Student 1: my variant is 26
Professor received an answer from Student 1:
    Student 1: my answer is ANSWER 8
```

Параметрами для исполняемого файла p.out являются:

- 1) N – количество студентов, планирующих сдавать экзамен

```
mastavtsev@mastavtsev-VirtualBox:~/IHT_2/7_points$ ./s.out
Student 1 received a mark from professor:
    Your mark is 9!
mastavtsev@mastavtsev-VirtualBox:~/IHT_2/7_points$ ./s.out
Student 1 received a mark from professor:
    Your mark is 10!
```

Исполняемый файл s.out не имеет параметров.

### Общая схема решаемой задачи:

Главным отличием от приложений на 4-6 баллов является то, что в данном, и в последующих приложениях, присутствуют две программы.

professor.c (p.out) – программа профессора, которая имеет такую же как и ранее модель общения с другими процессами (студентами), только теперь они не являются дочерними процессами.



student.c – приложение студента, модель общения такая же, как описано ранее

Если описать принцип работы программы в рамках задачи, то условно профессор обозначает, сколько студентов он готов принять на защите/экзамене и ждёт получения от процессов студентов, пока их не пройдёт  $n$  штук. Процессы студентов могут запускаться поотдельности, как показано на скриншоте выше, или же могут быть запущены при помощи скринта.

```
mastavtsev@mastavtsev-VirtualBox:~/IHT_2/7_points$ ./2_students.sh
Student 1 received a mark from professor:
    Your mark is 1!
Student 1 received a mark from professor:
    Your mark is 5!
```

**Важно:** процесс профессора всегда, в данном приложении и далее, должен быть запущен первым. Это необходимо для корректной инициализации и создания семафоров и разделяемой памяти, а также дальнейшего освобождения ресурсов. В рамках задачи, можем сказать так, что профессору необходимо первому войти в аудиторию и разложить в ней билеты с вопросам. Только после этого студенты имеют право входить)

## **8 баллов (семафоры System V)**

### **Файлы и каталоги:**

professor.c – программа профессора

student.c – программа студента

p.out – готовый исполняемый файл для профессора

s.out – готовый исполняемый файл для студента

2\_students.sh – скрипт для запуска двух студентов

### **Ввод и вывод данных:**

Аналогично задаче на 7 баллов.

### **Общая схема решаемой задачи:**

Отличием данной программы является использование семафоров System V, а также инструментов разделяемой памяти System V. Остальное взаимодействие процессов аналогично приложению на 7 баллов.

## **9 баллов (семафоры System V и очереди)**

### **Файлы и каталоги:**

professor.c – программа профессора

student.c – программа студента

p.out – готовый исполняемый файл для профессора

s.out – готовый исполняемый файл для студента

2\_students.sh – скрипт для запуска двух студентов

### **Ввод и вывод данных:**

Аналогично задаче на 7 и 8 баллов.

### **Общая схема решаемой задачи:**

Отличие от программы на 8 баллов заключается в использовании двух FIFO очередей, вместо разделяемой памяти System V.

## **10 баллов (семафоры POSIX и очередь)**

### **Файлы и каталоги:**

professor.c – программа профессора

student.c – программа студента

p.out – готовый исполняемый файл для профессора

s.out – готовый исполняемый файл для студента

2\_students.sh – скрипт для запуска двух студентов

### **Ввод и вывод данных:**

Аналогично задаче на 7-9 баллов.

### **Общая схема решаемой задачи:**

Отличие от других программ на 7-9 баллов в использовании комбинации семафоров POSIX и двух FIFO очередей.