

Цель работы

Цель лабораторной работы – изучение примитивов синхронизации и методов работы с ними, решение классической задачи узкого моста и тестирование решения в рамках операционной системы Pintos.

1 ХОД РАБОТЫ

1.1 Изменения в системе планирования Pintos

Были созданы четыре семафора для регулирования движения автомобилей всех видов по мосту: обычных автомобилей, движущихся вправо; обычных автомобилей, движущихся влево; автомобилей скорой помощи, движущихся вправо; автомобилей скорой помощи, движущихся влево. С помощью функции семафора `sema_down()` ограничивается доступ к мосту для автомобилей, а функция `sema_up()` открывает путь на мост.

```
struct semaphore semaphore_left_normal;  
struct semaphore semaphore_right_normal;  
struct semaphore semaphore_left_ambulance;  
struct semaphore semaphore_right_ambulance;
```

Одно из условий- нахождение на мосту не более 2-х машин, движущихся в одном направлении. Был создан счетчик машин на мосту и переменная направления движения на мосту в данный момент времени.

Были созданы переменные для подсчета всех автомобилей, желающих проехать по мосту, по виду. Данные счетчики позволяют контролировать условие: автомобили скорой помощи должны проезжать по мосту вне очереди. Таким образом, пока таких автомобилей не ноль, они запускаются на мост, пока обычные автомобили ждут своей очереди.

Инициализация всех глобальных переменных, в том числе и семафоров:

```

void narrow_bridge_init(void)
{
    max_count_threads = 2;
    flag_first_threads = 0;
    count_threads_on_bridge = 0;
    direction_move = 1; // 1 - left; 2 - right
    //global_flag_reverse_move = 0; // 1 - left; 2 - right
    all_threads_cars = 0;

    // i: 0 - left; 1 - right
    // j: 0 - normal car; 1 - ambulance
    for (int i = 0; i < 2; i++){
        for (int j = 0; j < 2; j++){
            numbers_threads[i][j] = 0;
        }
    }

    sema_init(&semaphore_left_normal, 0);
    sema_init(&semaphore_right_normal, 0);
    sema_init(&semaphore_left_ambulance, 0);
    sema_init(&semaphore_right_ambulance, 0);
}

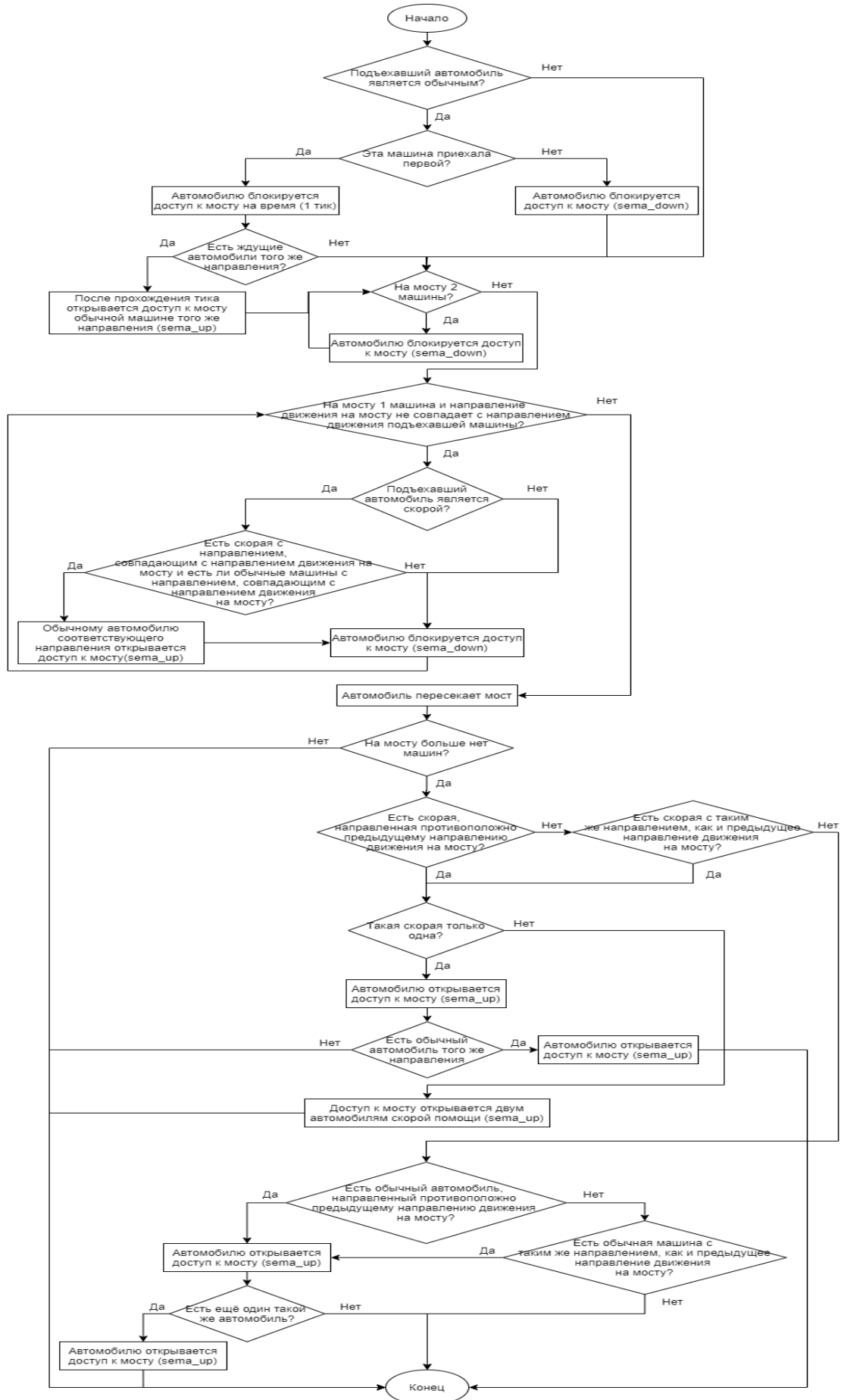
```

Реализована функция прибытия автомобилей к мосту `arrive_bridge()`. Если прибывший автомобиль является обычным, то ему блокируется доступ к мосту, чтобы первым делом пересекли мост автомобили скорой помощи. Однако первые две машины, если у них одинаковое направление или только одна, проезжают сразу, так как они первые, и никаких скорых еще в очереди не стоит.

Пока на мосту две машины, автомобили записываются в соответствующий список автомобилей, ждущих проезда, также к соответствующему счетчику ждущих прибавляется 1. Также в этой функции ведется подсчет автомобилей, находящихся на мосту, и запоминается направление движения на нем.

После была реализована функция выезда с моста `exit_bridge()`. При выезде автомобиля с моста из счетчика автомобилей, находящихся на мосту, вычитается 1. Если на мосту нет машин, то выбирается, какой машине откроется доступ к мосту. При каждом открытии пути к мосту из счетчиков автомобилей, ждущих у въезда на мост, вычитается единица.

1.2 Диаграмма взаимодействия процессов



2 РЕЗУЛЬТАТЫ ТЕСТОВ

```
Executing 'narrow-bridge 4 3 4 3':
(narrow-bridge) begin
(narrow-bridge) Vehicle:      1, prio: norm, direct: l -> r, ticks= 36
(narrow-bridge) Vehicle:      2, prio: norm, direct: l -> r, ticks= 36
(narrow-bridge) Vehicle:      8, prio: emer, direct: l -> r, ticks= 46
(narrow-bridge) Vehicle:      9, prio: emer, direct: l -> r, ticks= 46
(narrow-bridge) Vehicle:     12, prio: emer, direct: l <- r, ticks= 56
(narrow-bridge) Vehicle:     13, prio: emer, direct: l <- r, ticks= 56
(narrow-bridge) Vehicle:     10, prio: emer, direct: l -> r, ticks= 66
(narrow-bridge) Vehicle:     11, prio: emer, direct: l -> r, ticks= 66
(narrow-bridge) Vehicle:     14, prio: emer, direct: l <- r, ticks= 76
(narrow-bridge) Vehicle:      5, prio: norm, direct: l <- r, ticks= 76
(narrow-bridge) Vehicle:      3, prio: norm, direct: l -> r, ticks= 86
(narrow-bridge) Vehicle:      4, prio: norm, direct: l -> r, ticks= 86
(narrow-bridge) Vehicle:      6, prio: norm, direct: l <- r, ticks= 96
(narrow-bridge) Vehicle:      7, prio: norm, direct: l <- r, ticks= 96
(narrow-bridge) end
Execution of 'narrow-bridge 4 3 4 3' complete.
Timer: 106 ticks
Thread: 0 idle ticks, 106 kernel ticks, 0 user ticks
Console: 1390 characters output
Keyboard: 0 keys pressed
Powering off...
god@god-VirtualBox:~/Desktop/pintos/src/threads$
```

```
Calibrating timer... 427,622,400 loops/s.
Boot complete.
Executing 'narrow-bridge 1 3 4 10':
(narrow-bridge) begin
(narrow-bridge) Vehicle:      1, prio: norm, direct: l -> r, ticks= 48
(narrow-bridge) Vehicle:      5, prio: emer, direct: l -> r, ticks= 48
(narrow-bridge) Vehicle:      9, prio: emer, direct: l <- r, ticks= 58
(narrow-bridge) Vehicle:     10, prio: emer, direct: l <- r, ticks= 58
(narrow-bridge) Vehicle:     11, prio: emer, direct: l <- r, ticks= 68
(narrow-bridge) Vehicle:     12, prio: emer, direct: l <- r, ticks= 68
(narrow-bridge) Vehicle:     13, prio: emer, direct: l <- r, ticks= 78
(narrow-bridge) Vehicle:     14, prio: emer, direct: l <- r, ticks= 78
(narrow-bridge) Vehicle:     15, prio: emer, direct: l <- r, ticks= 88
(narrow-bridge) Vehicle:     16, prio: emer, direct: l <- r, ticks= 88
(narrow-bridge) Vehicle:      6, prio: emer, direct: l -> r, ticks= 98
(narrow-bridge) Vehicle:      7, prio: emer, direct: l -> r, ticks= 98
(narrow-bridge) Vehicle:     17, prio: emer, direct: l <- r, ticks= 108
(narrow-bridge) Vehicle:     18, prio: emer, direct: l <- r, ticks= 108
(narrow-bridge) Vehicle:      8, prio: emer, direct: l -> r, ticks= 118
(narrow-bridge) Vehicle:      2, prio: norm, direct: l <- r, ticks= 128
(narrow-bridge) Vehicle:      3, prio: norm, direct: l <- r, ticks= 128
(narrow-bridge) Vehicle:      4, prio: norm, direct: l <- r, ticks= 138
(narrow-bridge) end
Execution of 'narrow-bridge 1 3 4 10' complete.
```

```

367 pages available in user pool.
Calibrating timer... 419,020,800 loops/s.
Boot complete.
Executing 'narrow-bridge 20 0 0 21':
(narrow-bridge) begin
(narrow-bridge) Vehicle:      1, prio: norm, direct: l -> r, ticks= 33
(narrow-bridge) Vehicle:      2, prio: norm, direct: l -> r, ticks= 33
(narrow-bridge) Vehicle:     21, prio: emer, direct: l <- r, ticks= 43
(narrow-bridge) Vehicle:     22, prio: emer, direct: l <- r, ticks= 43
(narrow-bridge) Vehicle:     23, prio: emer, direct: l <- r, ticks= 53
(narrow-bridge) Vehicle:     24, prio: emer, direct: l <- r, ticks= 53
(narrow-bridge) Vehicle:     25, prio: emer, direct: l <- r, ticks= 63
(narrow-bridge) Vehicle:     26, prio: emer, direct: l <- r, ticks= 63
(narrow-bridge) Vehicle:     27, prio: emer, direct: l <- r, ticks= 73
(narrow-bridge) Vehicle:     28, prio: emer, direct: l <- r, ticks= 73
(narrow-bridge) Vehicle:     29, prio: emer, direct: l <- r, ticks= 83
(narrow-bridge) Vehicle:     30, prio: emer, direct: l <- r, ticks= 83
(narrow-bridge) Vehicle:     31, prio: emer, direct: l <- r, ticks= 93
(narrow-bridge) Vehicle:     32, prio: emer, direct: l <- r, ticks= 93
(narrow-bridge) Vehicle:     33, prio: emer, direct: l <- r, ticks= 103
(narrow-bridge) Vehicle:     34, prio: emer, direct: l <- r, ticks= 103
(narrow-bridge) Vehicle:     35, prio: emer, direct: l <- r, ticks= 113
(narrow-bridge) Vehicle:     36, prio: emer, direct: l <- r, ticks= 113
(narrow-bridge) Vehicle:     37, prio: emer, direct: l <- r, ticks= 123
(narrow-bridge) Vehicle:     38, prio: emer, direct: l <- r, ticks= 123
(narrow-bridge) Vehicle:     39, prio: emer, direct: l <- r, ticks= 133
(narrow-bridge) Vehicle:     40, prio: emer, direct: l <- r, ticks= 133
(narrow-bridge) Vehicle:     41, prio: emer, direct: l <- r, ticks= 143
(narrow-bridge) Vehicle:      3, prio: norm, direct: l -> r, ticks= 153
(narrow-bridge) Vehicle:      4, prio: norm, direct: l -> r, ticks= 153
(narrow-bridge) Vehicle:      5, prio: norm, direct: l -> r, ticks= 163
(narrow-bridge) Vehicle:      6, prio: norm, direct: l -> r, ticks= 163
(narrow-bridge) Vehicle:      7, prio: norm, direct: l -> r, ticks= 173
(narrow-bridge) Vehicle:      8, prio: norm, direct: l -> r, ticks= 173
(narrow-bridge) Vehicle:      9, prio: norm, direct: l -> r, ticks= 183
(narrow-bridge) Vehicle:     10, prio: norm, direct: l -> r, ticks= 183
(narrow-bridge) Vehicle:     11, prio: norm, direct: l -> r, ticks= 193
(narrow-bridge) Vehicle:     12, prio: norm, direct: l -> r, ticks= 193
(narrow-bridge) Vehicle:     13, prio: norm, direct: l -> r, ticks= 203
(narrow-bridge) Vehicle:     14, prio: norm, direct: l -> r, ticks= 203
(narrow-bridge) Vehicle:     15, prio: norm, direct: l -> r, ticks= 213
(narrow-bridge) Vehicle:     16, prio: norm, direct: l -> r, ticks= 213
(narrow-bridge) Vehicle:     17, prio: norm, direct: l -> r, ticks= 223
(narrow-bridge) Vehicle:     18, prio: norm, direct: l -> r, ticks= 223
(narrow-bridge) Vehicle:     19, prio: norm, direct: l -> r, ticks= 233
(narrow-bridge) Vehicle:     20, prio: norm, direct: l -> r, ticks= 233
(narrow-bridge) end
Execution of 'narrow-bridge 20 0 0 21' complete.

```

3 ДОПОЛНИТЕЛЬНОЕ ЗАДАНИЕ

Вариант 7. Спящий парикмахер(семафоры):

У парикмахера есть одно рабочее место и приемная с несколькими стульями. Когда у парикмахера нет посетителей — он спит в своем кресле. Если приходит посетитель — он будит парикмахера, он усаживает клиента в кресло и подстригает его. Когда парикмахер заканчивает подстригать клиента, он отпускает клиента и затем идет в приёмную, чтобы посмотреть, есть ли там

ожидающие клиенты. Если они есть, он приглашает одного из них и стрижет его. Если ждущих клиентов нет, он возвращается к своему креслу и спит в нем.

```
Pintos booting with 3,968 kB RAM...
367 pages available in kernel pool.
367 pages available in user pool.
Calibrating timer... 51,968,000 loops/s.
Boot complete.
Executing 'hair 5 100':
(hair) begin
(hair)
<<<<<hairstresser created>>>>>

(hair)
<<<<<client 1 created>>>>>

(hair) >>> client 1 go to barber >>> 134
(hair) >>> barber start cut >>> 134
(hair) >>> barber stop cut >>> 144
(hair)
<<<<<client 2 created>>>>>

(hair) >>> client 2 go to barber >>> 234
(hair) >>> barber start cut >>> 234
(hair) >>> barber stop cut >>> 244
(hair)
<<<<<client 3 created>>>>>

(hair) >>> client 3 go to barber >>> 334
(hair) >>> barber start cut >>> 334
(hair) >>> barber stop cut >>> 344
(hair)
<<<<<client 4 created>>>>>

(hair) >>> client 4 go to barber >>> 434
(hair) >>> barber start cut >>> 434
(hair) >>> barber stop cut >>> 444
(hair)
<<<<<client 5 created>>>>>

(hair) >>> client 5 go to barber >>> 534
(hair) >>> barber start cut >>> 534
(hair) >>> barber stop cut >>> 544
(hair) PASS
(hair) end
Execution of 'hair 5 100' complete.
Timer: 1034 ticks
Thread: 0 idle ticks, 1034 kernel ticks, 0 user ticks
Console: 1151 characters output
Keyboard: 0 keys pressed
```



```

Pintos booting with 3,968 kB RAM...
367 pages available in kernel pool.
367 pages available in user pool.
Calibrating timer... 209,510,400 loops/s.
Boot complete.
Executing 'hair 5 2':
(hair) begin
(hair)
<<<<<hairedresser created>>>>>

(hair)
<<<<<client 1 created>>>>>

(hair) >>> client 1 go to barber >>> 34
(hair) >>> barber start cut >>> 34
(hair)
<<<<<client 2 created>>>>>

(hair) >>> client 2 seat in chair >>> 36
(hair)
<<<<<client 3 created>>>>>

(hair) >>> client 3 seat in chair >>> 38
(hair)
<<<<<client 4 created>>>>>

(hair) >>> client 4 seat in chair >>> 40
(hair)
<<<<<client 5 created>>>>>

(hair) >>> client 5 seat in chair >>> 42
(hair) >>> barber stop cut >>> 44
(hair) >>> barber start cut >>> 44
(hair) >>> barber stop cut >>> 54
(hair) >>> barber start cut >>> 54
(hair) >>> barber stop cut >>> 64
(hair) >>> barber start cut >>> 64
(hair) >>> barber stop cut >>> 74
(hair) >>> barber start cut >>> 74
(hair) >>> barber stop cut >>> 84
(hair) PASS
(hair) end
Execution of 'hair 5 2' complete.
Timer: 542 ticks
Thread: 0 idle ticks, 542 kernel ticks, 0 user ticks
Console: 1133 characters output
Keyboard: 0 keys pressed
Powering off...

```

```

Pintos booting with 3,968 kB RAM...
367 pages available in kernel pool.
367 pages available in user pool.
Calibrating timer... 432,537,600 loops/s.
Boot complete.
Executing 'hair 5 4':
(hair) begin
(hair)
<<<<<hairdresser created>>>>>

(hair)
<<<<<client 1 created>>>>>

(hair) >>> client 1 go to barber >>> 56
(hair) >>> barber start cut >>> 56
(hair)
<<<<<client 2 created>>>>>

(hair) >>> client 2 seat in chair >>> 60
(hair)
<<<<<client 3 created>>>>>

(hair) >>> client 3 seat in chair >>> 64
(hair) >>> barber stop cut >>> 66
(hair) >>> barber start cut >>> 66
(hair)
<<<<<client 4 created>>>>>

(hair) >>> client 4 seat in chair >>> 68
(hair)
<<<<<client 5 created>>>>>

(hair) >>> client 5 seat in chair >>> 72
(hair) >>> barber stop cut >>> 76
(hair) >>> barber start cut >>> 76
(hair) >>> barber stop cut >>> 86
(hair) >>> barber start cut >>> 86
(hair) >>> barber stop cut >>> 96
(hair) >>> barber start cut >>> 96
(hair) >>> barber stop cut >>> 106
(hair) PASS
(hair) end
Execution of 'hair 5 4' complete.
Timer: 572 ticks
Thread: 0 idle ticks, 572 kernel ticks, 0 user ticks
Console: 1134 characters output
Keyboard: 0 keys pressed
Powering off...

```

4 ВЫВОД

Были изучены примитивы синхронизации и методы работы с ними, решена классическая задача узкого моста и протестировано решение в рамках операционной системы Pintos.