

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Образовательная программа бакалавриата «Программная инженерия»

Архитектура вычислительных систем
Пояснительная записка к микропроекту по теме «Многопоточное
программирование. Взаимодействие потоков.»
Вариант 24 (2)

Петров Иван Андреевич, БПИ195

Содержание

1	Текст задания	1
2	Описание используемых переменных	1
3	Сценарий взаимодействия объектов программы	1
4	Формат входных данных	2
5	Область допустимых значений входных данных	2
6	Тестовые примеры	2
6.1	Пример 1	2
6.2	Пример 2	2
6.3	Пример 3	3
6.4	Пример 4. Некорректные данные	4
7	Список используемых источников	4

1 Текст задания

Задача о Винни-Пухе или правильные пчелы. В одном лесу живут n пчел и один медведь, которые используют один горшок меда, вместимостью N глотков. Сначала горшок пустой. Пока горшок не наполнится, медведь спит. Как только горшок заполняется, медведь просыпается и съедает весь мед, после чего снова засыпает. Каждая пчела многократно собирает по одному глотку меда и кладет его в горшок. Пчела, которая приносит последнюю порцию меда, будит медведя. Создать многопоточное приложение, моделирующее поведение пчел и медведя.

2 Описание используемых переменных

- **pot** – текущее количество порций в горшке меда;
- **capacity** – вместимость горшка меда;
- **beesNumber** – количество пчел;
- **iterations** – количество итераций (пробуждений медведя);
- **mutex** – мьютекс для условных переменных;
- **bees** – условная переменная для блокировки потоков пчел;
- **bear** – условная переменная для блокировки потока медведя.

3 Сценарий взаимодействия объектов программы

В начале работы программы создаются **beesNumber** объектов потоков-производителей (пчел). Главный поток является потоком-потребителем (медведем). Когда горшок меда полон, поток-производитель блокируется условной переменной **bees**. Когда горшок меда не полон, поток-потребитель блокируется условной переменной **bear**. Для защиты операций чтения и записи в переменную **pot** используется двоичный семафор (мьютекс). После того как медведь проснулся **iterations** раз, работа программы завершается.

Пчелы. Перед началом работы с переменной **pot** мьютекс блокируется. Каждая пчела в каждой итерации цикла проверяет равенство значений переменных **pot** и **capacity**, то есть заполненность горшка с медом. Пока горшок с медом полон, все потоки пчел блокируются. Если не полон, то значение переменной **pot** увеличивается на 1 – пчела приносит одну порцию меда. После окончания работы с переменной **pot** мьютекс разблокируется. Если в ходе итерации после пополнения горшка с медом значение переменной **pot** становится равным значению переменной **capacity**, то есть горшок меда становится полным - разблокируются все потоки, в данный момент заблокированные переменной **bear**, то есть просыпается медведь. В конце своей итерации поток пчелы засыпает на 2 секунды - время, спустя которое одна пчела соберет и положит очередную порцию меда.

Медведь. Перед началом работы с переменной **pot** мьютекс блокируется. Медведь в каждой итерации цикла проверяет равенство значений переменных **pot** и **capacity**, то есть заполненность горшка с медом. Пока горшок с медом не полон, поток медведя блокируется. Если полон, то значение переменной **pot** обнуляется - медведь съедает весь мед. После окончания работы с переменной **pot** мьютекс разблокируется. В конце своей итерации поток медведя разблокирует все потоки, в данный момент заблокированные переменной **bees**, то есть пчелы продолжают заполнять горшок с медом.

4 Формат входных данных

Входные данные задаются через аргументы командной строки следующим образом:

<путь к исполняемому файлу> <вместимость горшка с медом> <количество пчел> <количество итераций>

5 Область допустимых значений входных данных

Все значения входных данных должны лежать в отрезке [1; 65535].

6 Тестовые примеры

6.1 Пример 1

Результат работы программы со следующими входными данными:

- вместимость горшка с медом = 5
- количество пчел = 10
- количество итераций = 2

```
[0.0040] Bee 1 puts 1 sip of honey into the pot (1/5).
[0.0170] Bee 2 puts 1 sip of honey into the pot (2/5).
[0.0220] Bee 3 puts 1 sip of honey into the pot (3/5).
[0.0270] Bee 4 puts 1 sip of honey into the pot (4/5).
[0.0340] Bee 5 puts 1 sip of honey into the pot (5/5).
[0.0390] The pot is full. Bee 5 wakes up the bear.
[0.0450] The bear woke up and ate all the honey.
[0.0510] The bear fell asleep again. The bees begin to collect honey.
[0.0580] Bee 10 puts 1 sip of honey into the pot (1/5).
[0.0650] Bee 8 puts 1 sip of honey into the pot (2/5).
[0.0700] Bee 9 puts 1 sip of honey into the pot (3/5).
[0.0770] Bee 7 puts 1 sip of honey into the pot (4/5).
[0.0820] Bee 6 puts 1 sip of honey into the pot (5/5).
[0.0870] The pot is full. Bee 6 wakes up the bear.
[4.0680] The bear woke up and ate all the honey.
[4.0720] The bear fell asleep again. The bees begin to collect honey.
[4.0780] Bee 1 puts 1 sip of honey into the pot (1/5).
[4.0830] Bee 2 puts 1 sip of honey into the pot (2/5).
[4.0880] Bee 6 puts 1 sip of honey into the pot (3/5).
[4.0950] Bee 3 puts 1 sip of honey into the pot (4/5).
[4.1020] Bee 5 puts 1 sip of honey into the pot (5/5).
[4.1080] The pot is full. Bee 5 wakes up the bear.
Number of iterations exceeded maximum.
```

6.2 Пример 2

Результат работы программы со следующими входными данными:

- вместимость горшка с медом = 5

- количество пчел = 2
- количество итераций = 2

```
[0.0000] The bear is sleeping until the pot is full...
[0.0050] Bee 1 puts 1 sip of honey into the pot (1/5).
[0.0100] Bee 2 puts 1 sip of honey into the pot (2/5).
[2.0150] Bee 1 puts 1 sip of honey into the pot (3/5).
[2.0310] Bee 2 puts 1 sip of honey into the pot (4/5).
[4.0260] Bee 1 puts 1 sip of honey into the pot (5/5).
[4.0310] The pot is full. Bee 1 wakes up the bear.
[4.0340] The bear woke up and ate all the honey.
[4.0380] The bear fell asleep again. The bees begin to collect honey.
[4.0430] Bee 2 puts 1 sip of honey into the pot (1/5).
[6.0470] Bee 2 puts 1 sip of honey into the pot (2/5).
[6.0520] Bee 1 puts 1 sip of honey into the pot (3/5).
[8.0550] Bee 2 puts 1 sip of honey into the pot (4/5).
[8.0600] The bear is sleeping until the pot is full...
[8.0710] Bee 1 puts 1 sip of honey into the pot (5/5).
[8.0760] The pot is full. Bee 1 wakes up the bear.
[8.0790] The bear woke up and ate all the honey.
[8.0830] The bear fell asleep again. The bees begin to collect honey.
[10.0720] Bee 2 puts 1 sip of honey into the pot (1/5).
[10.0870] Bee 1 puts 1 sip of honey into the pot (2/5).
[12.0920] Bee 2 puts 1 sip of honey into the pot (3/5).
Number of iterations exceeded maximum.
```

6.3 Пример 3

Результат работы программы со следующими входными данными:

- вместимость горшка с медом = 5
- количество пчел = 65535
- количество итераций = 2

```
[0.0020] Bee 1 puts 1 sip of honey into the pot (1/5).
[0.0090] Bee 2 puts 1 sip of honey into the pot (2/5).
[0.0160] Bee 3 puts 1 sip of honey into the pot (3/5).
[0.0230] Bee 4 puts 1 sip of honey into the pot (4/5).
[0.0300] Bee 5 puts 1 sip of honey into the pot (5/5).
[0.0380] The pot is full. Bee 5 wakes up the bear.
[4.7980] The bear woke up and ate all the honey.
[4.8020] The bear fell asleep again. The bees begin to collect honey.
[4.8080] Bee 65534 puts 1 sip of honey into the pot (1/5).
[5.1380] Bee 65535 puts 1 sip of honey into the pot (2/5).
[5.1420] Bee 448 puts 1 sip of honey into the pot (3/5).
[5.1470] Bee 449 puts 1 sip of honey into the pot (4/5).
[5.1530] Bee 447 puts 1 sip of honey into the pot (5/5).
[5.1580] The pot is full. Bee 447 wakes up the bear.
[9.1250] The bear woke up and ate all the honey.
[9.1290] The bear fell asleep again. The bees begin to collect honey.
[9.4430] Bee 393 puts 1 sip of honey into the pot (1/5).
[9.4490] Bee 57712 puts 1 sip of honey into the pot (2/5).
[9.4540] Bee 248 puts 1 sip of honey into the pot (3/5).
[9.4600] Bee 249 puts 1 sip of honey into the pot (4/5).
[9.4650] Bee 250 puts 1 sip of honey into the pot (5/5).
[9.4690] The pot is full. Bee 250 wakes up the bear.
Number of iterations exceeded maximum.
```

6.4 Пример 4. Некорректные данные

При вводе некорректных данных программа выводит сообщение, содержащее информацию о требуемом формате данных.

```
Usage: <path to executable file> <capacity of honey pot 1-65535> <number of bees 1-65535> <number of iterations 1-65535>
```

7 Список используемых источников

1. Легалов А.И. «Архитектура ВС. Параллельные ВС. Многопоточность». - Режим доступа: <http://softcraft.ru/edu/comparch/lect/07-parthread/>
2. Легалов А.И. «Многопоточность. Простая многопоточная программа. Основные функции». - Режим доступа: <http://softcraft.ru/edu/comparch/practice/thread/01-simple/>
3. Легалов А.И. «Многопоточность. Синхронизация потоков. Методы синхронизации». - Режим доступа: <http://softcraft.ru/edu/comparch/practice/thread/02-sync/>
4. Легалов А.И. «Параллельное программирование. Взаимодействие и синхронизация. Итоговое задание». - Режим доступа: <http://softcraft.ru/edu/comparch/tasks/mp02/>
5. POSIX Threads Programming. - Режим доступа: <https://computing.llnl.gov/tutorials/pthreads/>
6. Википедия. POSIX Threads. - Режим доступа: https://ru.wikipedia.org/wiki/POSIX_Threads