

Задача о больнице. В больнице два дежурных врача принимают пациентов, выслушивают их жалобы и отправляют или к стоматологу, или к хирургу, или к терапевту. Стоматолог, хирург и терапевт лечат пациентов. Каждый врач может принять только одного пациента за раз. Пациенты стоят в очереди к врачам и никогда их не покидают. Создать многопоточное приложение, моделирующее рабочий день клиники.

4 балла:

- 1) Приведено условие задачи.
- 2) Под данное условие наиболее подходящей моделью будет “Управляющий и рабочие”, а если быть точнее “Управляющие и рабочие”. Два главных потока – два дежурных врача – обрабатывают входные данные и запускают действие рабочих потоков – стоматолога, хирурга и терапевта.
- 3) Входные жалобы – количество пациентов, жалобы каждого из них в одно слово. Если пациенту требуется стоматолог, вводится слово “tooth”, если хирург – “injury”, если нужен терапевт, то вводится любое другое слово, обозначающее жалобу.
- 4) Реализовано консольное приложение, код расположен в файле main.cpp. В качестве синхропримитива были выбраны семафоры.
- 5) Ввод данных из консоли доступен если при вводе в командную строку не передается дополнительных параметров.

5 баллов:

- 1) Комментарии, поясняющие выполняемые действия и использование переменных присутствуют в файле с кодом.
- 2) Сценарий: в больнице есть очередь пациентов. Их параллельно обрабатывают два дежурных врача, каждого нового пациента будет обслуживать первый освободившийся врач. Каким образом дежурный врач обслуживает людей: он выслушивает жалобы пациента и из этих данных делает вывод, врач какой специальности нужен. Врачи (стоматолог, хирург, терапевт) независимы друг от друга и могут принимать пациентов одновременно. Если же врач нужной специальности занят, то дежурный врач ждет, пока он освободится и только потом перейдет к обслуживанию нового пациента.

6 баллов:

- 1) Очередь пациентов построена как очередь (queue) строк, в которых указаны жалобы каждого человека. Дежурные врачи реализованы как два потока, обращающиеся к очереди. Для ограничения доступа к этой очереди (исключение ситуации взятия потоками одного и того же пациента и удаления лишнего) используется семафор que. Во время взятия нового пациента, очередь блокируется с помощью sem\_wait(que), и возобновляет свою работу после удаления пациента из очереди с помощью sem\_post(que). Для индикатора занятости каждого из врачей были созданы булевы переменные dentBusy, surgBusy, therBusy. Прием врача реализован в виде потока, запускаемого дежурным врачом. Порядок его действий: изменить соответствующую переменную занятости, имитация приема (задержка времени usleep), вывод результатов на экран, изменение соответствующей переменной занятости. Если врач занят, а дежурный врач должен отправить к нему нового пациента, то дежурный ожидает, пока врач не освободится с помощью while (соответствующая переменная занятости). Для более эффективного решения проблемы с одновременным приемом пациентов на переменные занятости были заведены семафоры dent, surg, ther. В таком случае, если два пациента все-таки пробрались к одному врачу одновременно, то данные семафоры ограничивают доступ к переменным занятости, что позволяет проводить прием

только для одного пациента (два одновременно запущенных потока превращаются в два последовательных).

- 2) Предусмотрен ввод данных из командной строки

7 баллов:

- 1) Чтобы вводить данные из файла, при запуске программы из командной строки необходимо указать ровно 1 параметр, имя файла, из которого будет происходить чтение. Вывод информации происходит как в консоль, так и в прилагающийся файл output.txt
- 2) Произведено расширение ввода данных в связи с вышеуказанными изменениями
- 3) input.txt – файл с входными параметрами, output.txt – файл с выходными данными