Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования   
«Алтайский государственный техничеcкий университет им. И. И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

Отчет защищен с оценкой \_\_\_\_\_

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Отчет

по лабораторной работе № 8

по дисциплине «Операционные системы»

Студент гр. ПИ-82

Иванков Д.В.

Преподаватель доцент, к. т. н.

Боровцов Е. Г.

Барнаул 2020

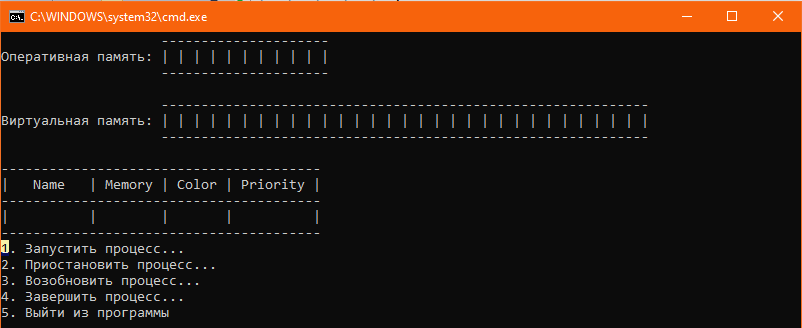
В данной лабораторной работе нужно было реализовать алгоритмы управления памятью и swapping.

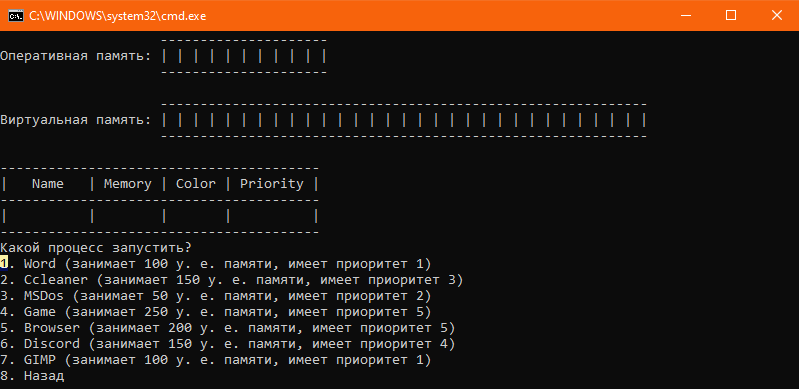
Я реализовал управление памятью в моделированной системе с приоритетным планированием процессов.

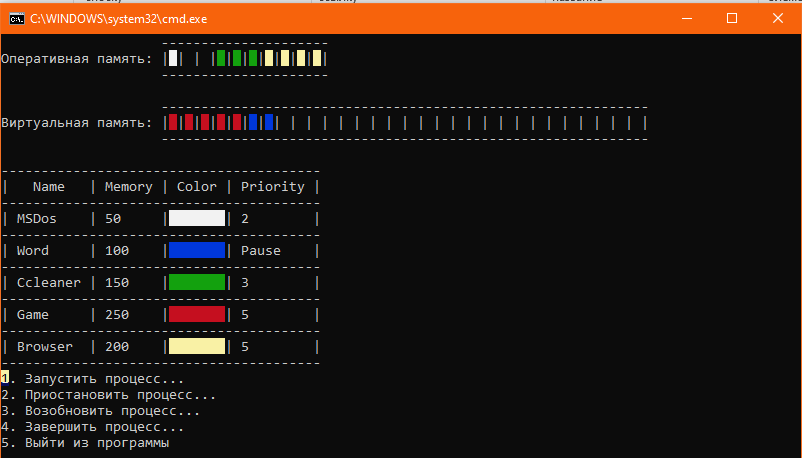
Я использовал языки c/c++, написал консольное приложение в духе MS-DOS.

**Работа программы**

Процессы, их название, занимаемая ими память, и их приоритет хранятся в программе в контейнере vector. Чтобы не путать их между собой, я каждому присвоил свой цвет. Количество оперативной и виртуальной памяти я задал константами 500 и 1500 соответственно. Чтобы красиво выводить всё это дело, я фрагментировал память блоками по 50 у. е памяти. Следовательно, занимаемая процессами память так же должна быть кратна 50. Так же для корректной работы приложения, процессы не должны иметь одинаковых имён. Я добавил 7 процессов с индивидуальными цветами, чтобы не запутаться. При желании можно добавить процесс, но цветов в консоли маловато и у каждого свой код. Можно добавить процесс с повторяющимся цветом, но тогда могут возникнуть ситуации, когда сложно будет различить что произошло в программе. Все запущенные процессы выводятся в таблицу со столбцами: название, память, цвет, приоритет. У мною добавленных процессов, максимальный приоритет 5, однако при желании можно выставить больший приоритет. Минимальный приоритет 0. Специальные значения приоритета -1 и -2 отвечают за промежутки и приостановленные процессы соответственно. Для оперативной и виртуальной памяти я вывожу прогресс-бары, в которых указываются запущенные процессы. Для взаимодействия с программой я сделал меню, в котором представлен весь функционал: добавление процесса, удаление процесса, приостановка процесса и возобновление процесса. При выборе одно из пунктов меню, открывается подменю, в котором можно выбрать над каким процессов произвести операцию или вернуться в меню. Для позиционирования в меню я выделяю номер пункта жёлтым фоном.



После любого действия над процессами, меню перерисовывается, учитывая изменения.



Функционал

1. Добавление процесса.

Процессы добавляются в оперативную память, если есть место. Если места нет, процесс помещается в виртуальную память и ждёт свою очередь (об этом позже).

1. Удаление процесса.

Процесс удаляется, где бы он ни находится (в оперативной или виртуальной памяти) и освобождает место. Если он был в оперативной памяти, то в ней освободилось место, поэтому автоматически запускается процесс своппинга (подкачки страниц другого процесса). О нём позже. Есть возможность удалить все процессы. При этом очищается оперативная и виртуальная память.

1. Приостановка процесса.

Процессу выдаётся специальный приоритет (-2), говорящий о том, что процесс приостановлен. Если процесс находится в оперативной памяти, то он скидывается в виртуальную память и автоматически запускается своппинг. Если процесс находится в виртуальной памяти, то он там и остаётся.

1. Возобновление процесса.

Процесс однозначно находится в виртуальной памяти, потому что при приостановке процесс помещается туда. Процессу присваивается приоритет, который был у него до приостановки. Если он помещается в оперативную память, то производится добавление данного процесса с удалением его из виртуальной памяти. Если нет, то он остаётся в виртуальной памяти ждать своей очереди.

1. Своппинг и система приоритетов.

Он происходит автоматически, при освобождении памяти в оперативке, при условии, что виртуальная память не пустая. Сначала находится максимальный приоритет у процессов, находящихся в виртуальной памяти. Затем, происходит попытка добавить процесс с максимальным приоритетом. Если после его добавления осталась память, то берётся следующий по приоритетности процесс и идёт попытка добавить его. И так далее, пока не проверятся все процессы. Таким образом происходит оптимизация памяти, не самая эффективная, но и не самая плохая. Важно отметить, что присутствует система “поощрений и штрафов”, а именно, если в оперативную память при своппинге помещается процесс с максимальным приоритетом, то его приоритет уменьшается на один. При этом, если процессы с минимальным приоритетом могли добавиться в оперативную память, но из-за высокоприоритетных процессов не были добавлены, так как не хватило места, то их приоритет повышается на один. Если максимальный и минимальный приоритеты равны, но приоритеты процессов не изменяются. Это сделано, чтобы высокоприоритетные процессы не выполнялись постоянно, а низкоприоритетные не ждали своей очереди бесконечно. Изначально я реализовал по-другому: у любого процесса, добавляемого при своппинге в операционную память, понижался приоритет на один, однако я заметил, что при этом иногда страдают процессы с низким приоритетом, поэтому решил “штрафовать” высокоприоритетные, и “поощрять” низкоприоритетные.

Управление меню состоит из трёх клавиш: стрелка вниз, стрелка вверх – для перемещения и enter – для выбора пункта меню.