

Цель работы:

Целью выполнения лабораторной работы является сравнение двух лабораторных установок, моделирующих диспетчер задач ОС.

Общие впечатления:

Между лабораторными установками, старой и новой, есть некоторые основные различия, которые сложно отнести к плюсам или минусам, поэтому проще высказать свое мнение в целом.

Основные различия между установками – это выбранный в установках «подход» к прохождению заданий и способ осуществления проверки результатов шага.

Сначала проще рассмотреть проверку результатов шага. В старой установке от пользователя требовалось осуществить действия по добавлению в очередь, извлечение из очереди и переключению на тот же самый процесс после истечению кванта времени или передачи управления системе, когда процесс все равно оставался выполняться из-за более высокого по сравнению с остальными приоритета. Довольно быстро к этому привыкаешь, но все равно каждый раз слегка раздражает это делать. В новой установке можно сразу нажать «Подтвердить». Это удобно и экономит время, а акцент на том, что у нас заново выбирается этот же процесс, можно сделать в методическом пособии и спрашивать об этом на защитах.

Теперь же можно рассмотреть, в чем заключается «подход» к прохождению заданий в каждой установке.

В старой лабораторной установке есть счетчик количества ошибок, и это очень сильно раздражает пользователя. С крайне высокой вероятностью он предпочтет пройти свой вариант минимум два раза: первый раз для обучения, а второй раз для получения хорошего результата с наименьшим количеством ошибок. Если же у пользователя разыграется перфекционизм, то он может пройти одно и то же задание больше двух раз. Ещё есть проблема в том, что более-менее «улавливаешь» суть задания уже на 10 шаге, а после 20 шага уверенно его проходишь, поэтому на самом деле можно было бы не проходить первый раз до конца, но пользоваться это, скорее всего, сделает, чтобы проверить задание полностью на наличие каких-либо «ловушек».

Что же получается в старой установке? Пользователь закончил «тестовое» прохождение и начал проходить свой вариант заново. Несмотря на то, что уже все примерно известно, пользователь должен быть максимально внимательным,

чтобы не допустить случайных ошибок, которые во время второго прохождения ему видеть совсем не хочется. Но из-за большого количества заданий, при котором начинает играть роль их однообразие, а также начинает влиять накопившаяся после «тестового» прохождения утомленность, глаз может «замылиться», в результате чего ошибки все равно могут быть, что приведет к еще большему раздражению, т.е. сработает эффект снежного кома.

Тут можно провести аналогию со старыми играми, когда какой-нибудь уровень делали очень сложным, из-за чего игроку приходилось тратить много времени, чтобы улучшить свой результат и наконец-то пройти этот уровень. Если судить по году создания установки, то ее авторы, скорее всего, выросли на подобных играх, поэтому нет ничего удивительного, что они ими вдохновлялись в то время. К счастью, установка позволяет отменить шаг, чтобы его перепройти, и даже допускает возможность оставить шаг с ошибкой, если уж совсем не получается его решить. Установка вполне корректно учитывает выполнение шагов даже с ошибкой в одной из предыдущих, хотя так может нарушиться задумка варианта по созданию на определенном шаге интересной ситуации.

В новой лабораторной установке нет никакого счетчика ошибок, но зато перейти к следующему шагу можно только после корректного выполнения текущего. Если продолжать проводить аналогии, то это больше похоже на современные игры. Важно разобраться и пройти шаг правильно, но при этом нет лишнего напряжения и раздражения, которые возникали при прохождении заданий в старой установке. Таким образом, пользователь может уделить больше внимания сути задания, а не борьбе с раздражением и особенностями установки.

Главное отличие новой установки, которое было осознанно не упомянуто ранее, заключается в генерации заданий вместо готовых вариантов. Этим можно очень хорошо воспользоваться, если также учитывать возможность пользователя уделять больше внимания сути задания и ранее упомянутое суждение, что достаточно 10-20 шагов для начала уверенного прохождения задания. Если сформулировать кратко, то можно добавить генерацию таких вариантов, где одно задание будет включать в себя сразу несколько подзаданий с разными стратегиями выбора свободного блока памяти или дисциплинами планирования процессов. Каждое подзадание будет включать 10-15 шагов, но нужно будет доработать генерацию для таких вариантов (с подзаданиями в заданиях), чтобы точно возникали «интересные ситуации», подчеркивающие особенности дисциплин планирования. Очевидно, что пользователь может не сразу «переключиться» между разными дисциплинами, поэтому отсутствие счетчика ошибок в данном случае несомненно является плюсом.

Критика и предложения:

Генерация.

В случае диспетчеризации памяти генерация работает достаточно хорошо, потому что само задание в целом проще, чем остальные.

В случае диспетчеризации процессов генерацию все же еще стоит доработать, так как в одном задании она может создавать слишком много процессов, причем в основном с большим временем выполнения, и редко удалять лишь некоторые, а в другом задании процессы создаются и почти сразу же удаляются, из-за чего наличие в очереди хотя бы двух процессов уже может показаться чудом.

В идеале, наверное, стоит сбалансировать генерацию так, чтобы в одном задании было и много, и мало, и на каких-то шагах между этим средне. Если же добавлять возможность генерации вариантов с подзаданиями в заданиях, то там должно быть одновременное количество процессов не больше, чем «средне» в обычных вариантах, и крайне желателен какой-либо случай, типичный для дисциплины выполняемого подзадания.

The screenshot shows a window titled "Модель диспетчера задач ОС" (Model of OS Task Scheduler). It has tabs for "Задание" (Task) and "Справка" (Help). The "Диспетчеризация процессов" (Process Scheduling) tab is active, showing a strategy of "SRT". A status message says "Процесс PID = 0 выдал запрос на ввод/вывод" (Process PID = 0 issued a request for input/output). A progress indicator shows "Обработано заявок: 38 из 40" (38 of 40 requests processed). There are "Подтвердить" (Confirm) and "Сбросить" (Reset) buttons. Below the status bar is a table of processes and two FIFO queues.

PID	PPID	Состояние	Приоритет	Б. приоритет	t	T	T-t
8	-1	Q	0	0	6	4	-2
6	-1	W	0	0	6	5	-1
0	6	E	0	0	4	4	0
19	-1	W	0	0	12	22	10
4	8	Q	0	0	1	15	14
15	-1	Q	0	0	6	20	14
23	0	Q	0	0	0	16	16
1	0	Q	0	0	0	22	22
14	19	Q	0	0	0	22	22
16	15	Q	0	0	0	26	26
5	15	Q	0	0	0	28	28
18	-1	Q	0	0	0	29	29

Очереди FIFO

Queue 1	Queue 2
15	15
4	4
23	23
14	14
1	1
16	16
5	5
18	18
9	9
22	22
8	8

Рисунок 1 – Большое количество процессов

Проверка результата.

Хотя новая лабораторная установка гораздо дружелюбнее, лучше дополнительно уведомить пользователя о нюансах организации очереди в случае одинакового размера или времени выполнения.

В диспетчеризации процессов, как видно на рисунке 1, процессы с одинаковым временем выполнения должны быть отсортированы по убыванию PID (процессы 15 и 4, 14 и 1), иначе выдаст ошибку, как показано на рисунке 2.

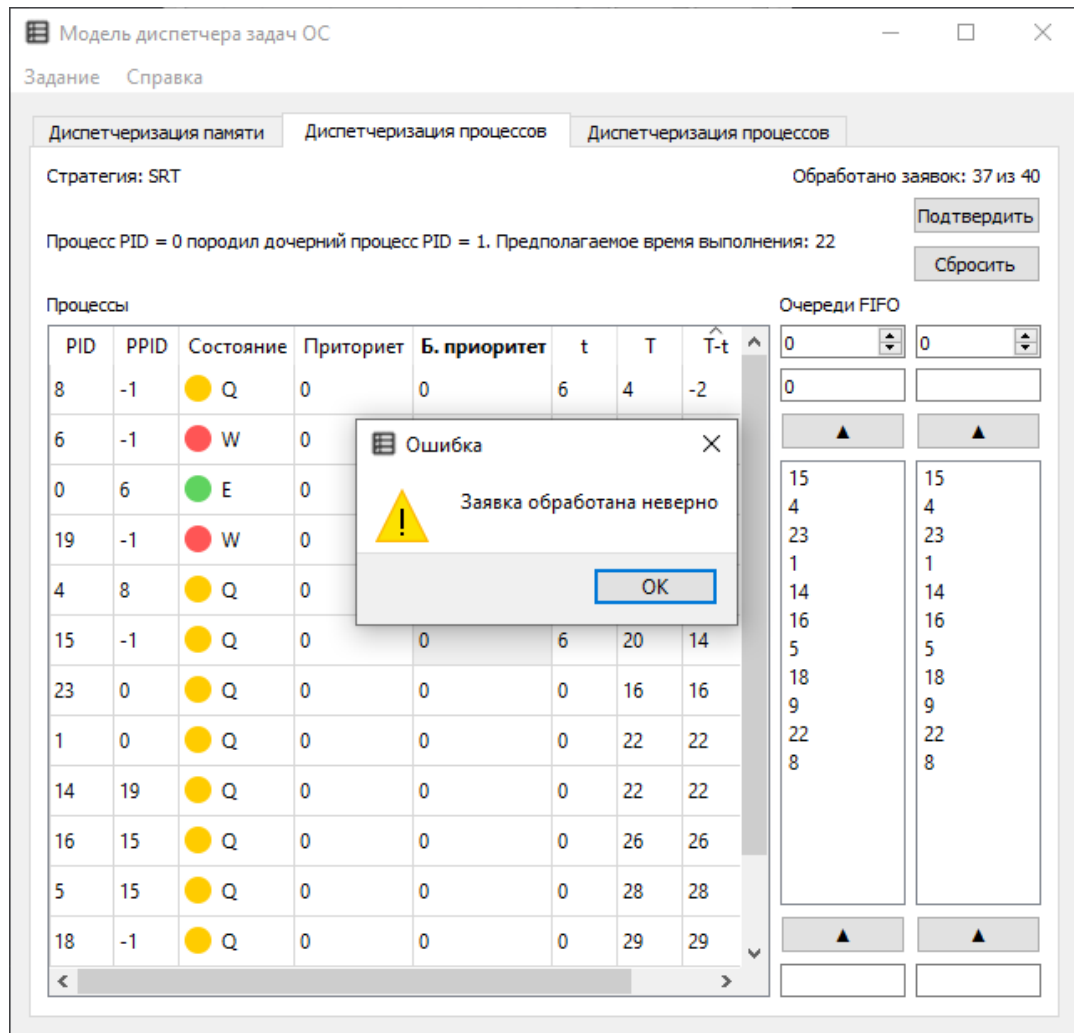


Рисунок 2 – Пример ошибки из-за сортировки в очереди

В диспетчеризации памяти свободные блоки одинакового размера еще должны быть отсортированы и по возрастанию адреса, иначе ответ не принимается.

На рисунке 3 представлено, что сначала должен быть блок с адресом 181, а затем блок с адресом 255.

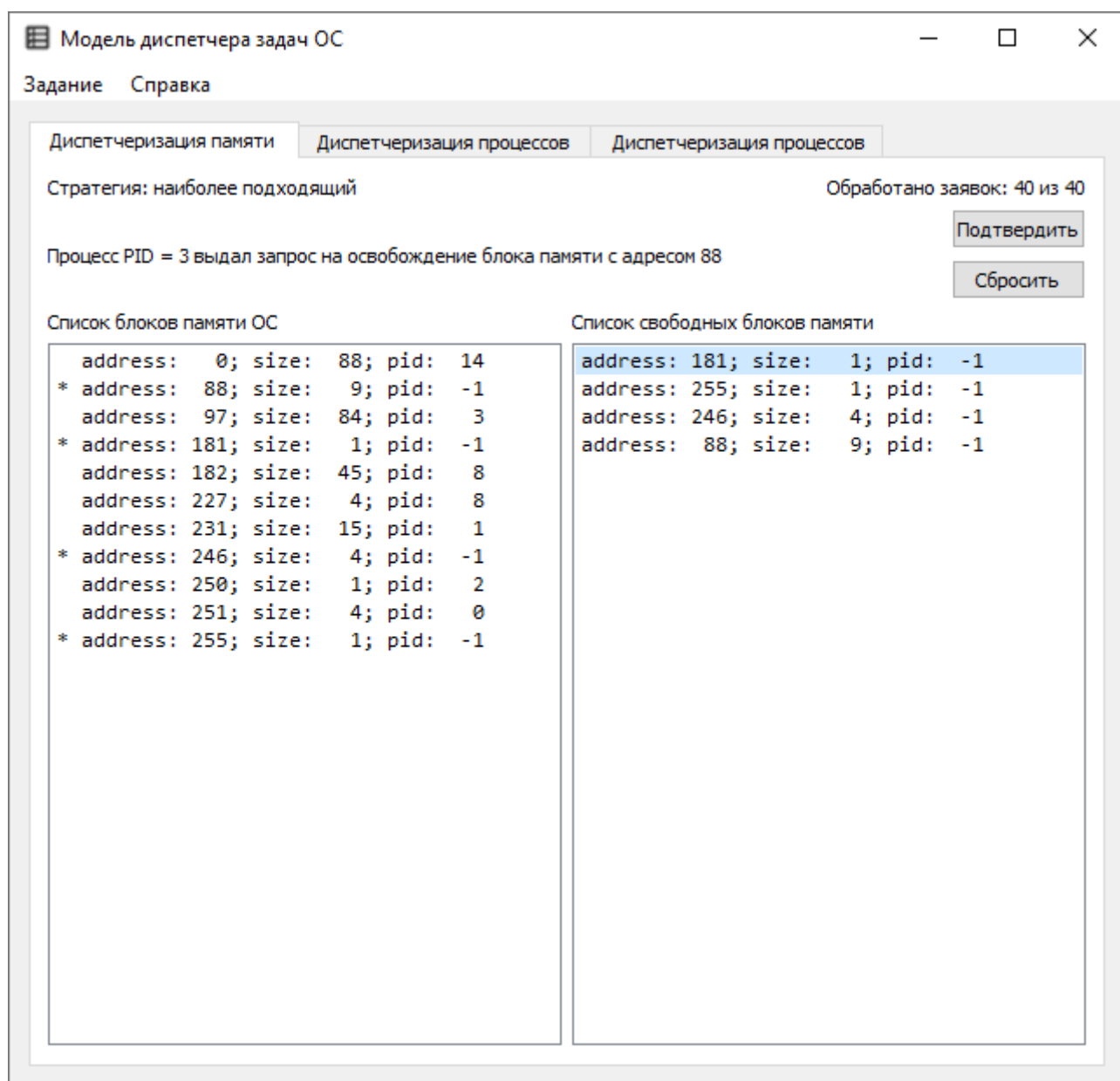


Рисунок 3 – Задание по диспетчеризации памяти

Оформление.

В старой лабораторной установке был больше шрифт у информации о произошедшем событии, что в какой-то степени снижало нагрузку на глаза. Переделывать размер шрифтов всего приложения нецелесообразно, но размер шрифта информации о событии, наверное, все же стоит увеличить.

Рассмотрим рисунок 3. Все же стоит выделить столбцы, хотя бы увеличив количество пробелов между ними, но, наверное, проще сделать табличкой, как в старой установке. Также выделение свободных блоков при помощи * в списке блоков памяти ОС менее наглядно, чем выделение цветом в старой установке. Отсутствует сумма размеров свободных блоков.

Интерфейс диспетчеризации процессов радует глаз больше, чем в старой установке, но нужно доработать окно добавления нового процесса (представлено на рисунке 4).

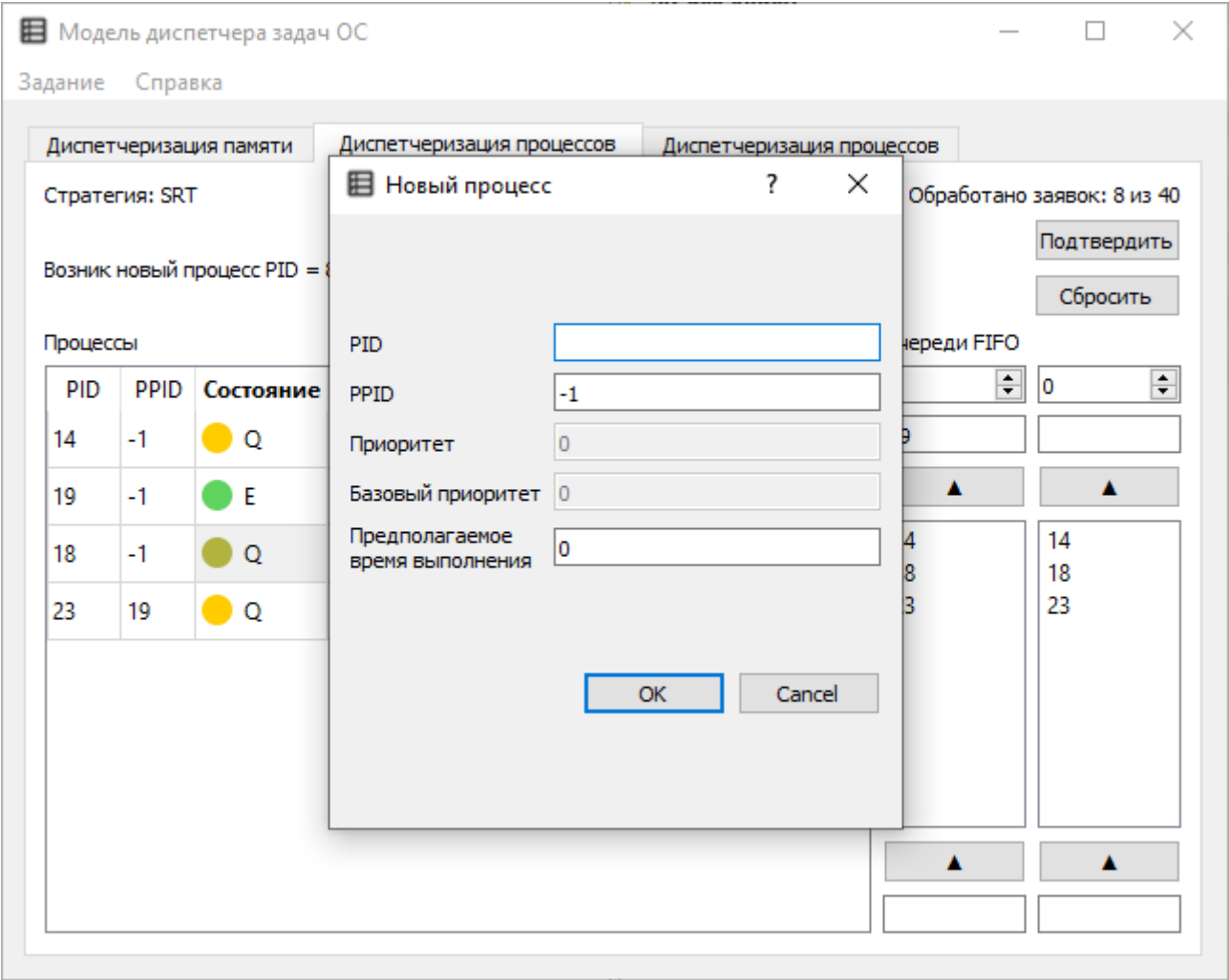


Рисунок 4 – Добавление нового процесса

Много свободного пространства, заслоняющего собой задание. Непереведенные кнопки подтверждения и отмены. Загадочная кнопка со знаком вопроса рядом с кнопкой закрытия окна, после нажатия на которую курсор превращается в знак запрета в пределах окна создания нового процесса и нужно просто нажать, чтобы курсор вернулся в нормальное состояние.

Еще, возможно, стоит добавить изменение ширины столбцов таблицы при увеличении размера основного окна, чтобы не было так много пустого места, как показано на рисунке 5.

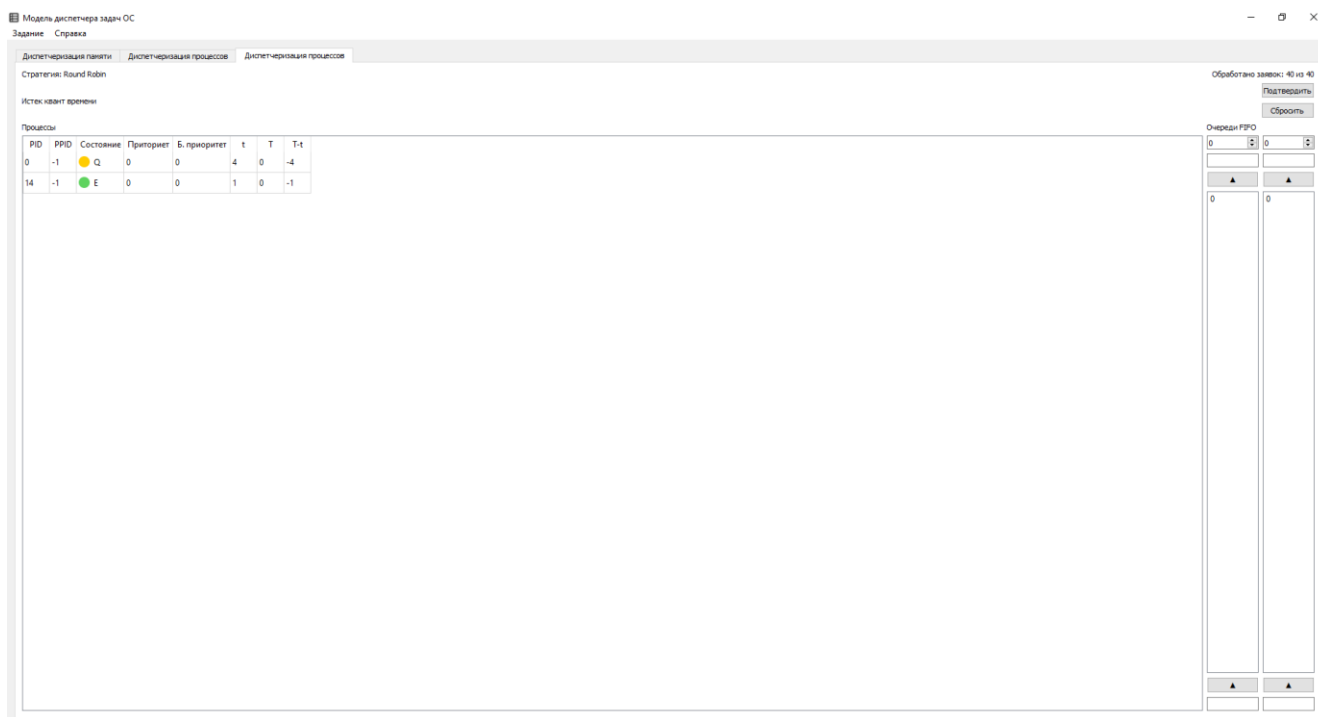


Рисунок 5 – Внешний вид при разворачивании окна на весь экран.

В заданиях, где применяются дисциплины с большим количеством очередей, отображение всех очередей сразу может смотреться не очень красиво, но стоит поэкспериментировать и, например, попробовать вариант с отображением по 4 очереди за раз и одновременным переключением между ними (отображены очереди с 0 по 3, нажимаем кнопку «Вверх» и сразу переходим на отображение очередей с 4 по 7).

Взаимодействие с интерфейсом.

Интересным вариантом улучшения программы является добавление горячих клавиш для действий (помимо нажатия ПКМ и выбора действия в меню добавить возможность нажать по процессу ЛКМ и, например, клавишу “W”, чтобы переключить его в состояние ожидания).

Можно сделать необычное сочетание клавиш, которое будет известно преподавателю, для вызова скрытого окна со статистикой, и можно будет добавить скрытый счетчик ошибок, если все же необходимо дополнительно оценить качество выполнения лабораторной работы студентом, но тогда нужно будет применять шифрование для файлов сохранений.

Вывод:

В ходе выполнения лабораторной работы были выполнены задания на обеих установках, благодаря чему были сформулированы критика и предложения по улучшению новой лабораторной установки.