Министерство образования и науки Российской Федерации

федеральное государственное бюджетное образовательное

учреждение высшего образования

«Алтайский государственный технический университет

им. И.И. Ползунова»

Факультет \_\_\_информационных технологий\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра \_\_\_\_прикладной математики\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Специальность (направление, профиль) \_\_ПИ\_\_\_\_\_\_\_\_\_\_\_\_\_

Курсовой проект

защищен с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ненайденко А.С.\_

(подпись руководителя проекта) (инициалы, фамилия)

“\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Курсовой проект

ТЕМА

(тема курсового проекта)

по дисциплине \_\_\_\_\_\_\_\_\_\_\_Операционные системы\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_КП 09.03.04.06.000 ПЗ\_\_\_\_\_\_\_

(обозначение документа)

Студент группы ПИ92 Шульпов\_В.М.\_\_\_\_\_\_\_\_\_25.05.2022

(фамилия, имя, отчество) (подпись) (дата)

Руководитель

проекта доцент, к.т.н.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Ненайденко А.С.

(должность, ученое звание) (подпись) (инициалы, фамилия)

БАРНАУЛ 2022

Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Алтайский государственный технический университет

имени И.И. Ползунова»

Факультет информационных технологий

Кафедра «Прикладная математика»

З А Д А Н И Е

на курсовой проект по дисциплине «Операционные системы»

студенту группы ПИ-92 Шульпову Виктору Максимовичу

Тема курсового проекта: «тема».

|  |  |
| --- | --- |
| Номер варианта | Задание |
| 4 | **Моделирование поведения мультизадачной системы (классического мультипрограммирования)** пакетного режима с переменным числом задач и дисциплиной приоритетного планирования с системой штрафов и поощрений. |

Обзор предметной области

**Мультипрограммирование** — способ организации выполнения нескольких программ на одном компьютере.

**Мультипрограммирование бывает**: в пакетных системах, [системах реального времени](https://ru.wikipedia.org/wiki/Система_реального_времени) и в системах разделения времени.

**Пакетное задание** ([англ.](https://ru.wikipedia.org/wiki/Английский_язык) batch job) — способ запуска задач на исполнение для избегания простоев вычислительных систем. Фактически представляет собой список запускаемых программ с указанием параметров запуска и входных данных. Задачи запускаются последовательно.

Задачи, планируемые к выполнению, называются пакетом. Переключение между задачами в пакетном режиме инициируется выполняющейся в данный момент задачей, поэтому промежутки времени выполнения той или иной задачи не определены.

**Пакетные системы:**

• производительность — выполнение максимального количества

заданий в час;

• оборотное время — минимизация времени между представлением

задачи и ее завершением;

• использование центрального процессора — поддержка постоянной

загруженности процессора

**Приоритетное планирование делится** на два основных типа:

* Упреждающее планирование

В упреждающем планировании задачи в основном назначаются с их приоритетами. Иногда важно запустить задачу с более высоким приоритетом перед другой задачей с более низким приоритетом, даже если задача с более низким приоритетом все еще выполняется. Задача с более низким приоритетом удерживается некоторое время и возобновляется, когда задача с более высоким приоритетом завершает свое выполнение.

* Непланирующее планирование

В этом типе метода планирования ЦП был выделен для определенного процесса. Процесс, который удерживает процессор занятым, освободит процессор либо переключением контекста, либо завершением. Это единственный метод, который можно использовать для различных аппаратных платформ. Это потому, что ему не нужно специальное оборудование (например, таймер), например, упреждающее планирование.

**Приоритет** – число, характеризующее степень привилегированности процесса при использовании ресурсов (целое, дробное, больше нуля, меньше нуля).

**Приоритетное планирование**

Каждому процессу присваивается значение приоритетности и запускается тот процесс, который находится в состоянии готовности и имеет наивысший приоритет. Чтобы предотвратить бесконечное выполнение высокоприоритетных процессов, планировщик должен понижать уровень приоритета текущего выполняемого процесса с каждым сигналом таймера (то есть с каждым его прерыванием). Если это действие приведет к тому, что его приоритет упадет ниже приоритета следующего по этому показателю процесса, произойдет переключение процессов. Возможна и другая альтернатива: каждому процессу может быть выделен максимальный квант допустимого времени выполнения. Когда квант времени будет исчерпан, шанс запуска будет предоставлен другому процессу, имеющему наивысший приоритет. Приоритеты могут присваиваться процессам в статическом или в динамическом режиме (система «поощрений и штрафов»)

**~~Переключение контекста~~**~~(~~[~~англ.~~](https://ru.wikipedia.org/wiki/Английский_язык)~~context switch) — в~~[~~многозадачных~~](https://ru.wikipedia.org/wiki/Многозадачность)[~~ОС~~](https://ru.wikipedia.org/wiki/Операционная_система)~~и средах - процесс прекращения выполнения процессором одной задачи (процесса, потока, нити) с сохранением всей необходимой информации и состояния, необходимых для последующего продолжения с прерванного места, и восстановления и загрузки состояния задачи, к выполнению которой переходит процессор.~~

## **~~Преимущества приоритетного планирования~~**

* ~~Простой в использовании метод планирования~~
* ~~Процессы выполняются на основе приоритета, поэтому высокому приоритету не нужно долго ждать, что экономит время~~
* ~~Этот метод обеспечивает хороший механизм, где можно точно определить относительную важность каждого процесса.~~
* ~~Подходит для приложений с меняющимися требованиями времени и ресурсов.~~

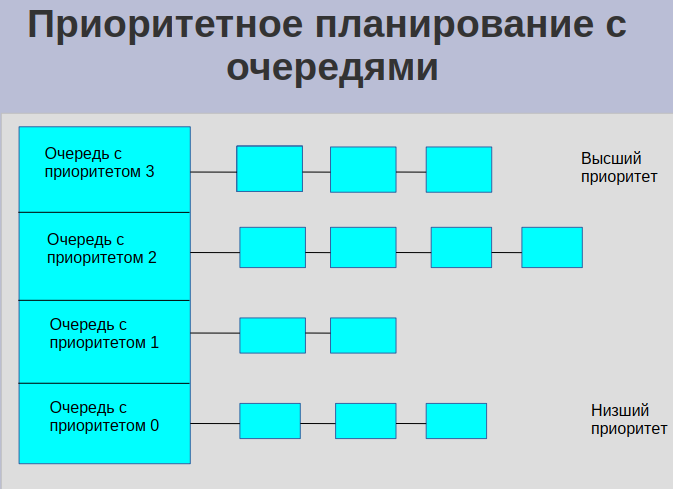
## **~~Недостатки приоритетного планирования~~**

* ~~В случае сбоя системы все процессы с низким приоритетом теряются.~~
* ~~Если процессы с высоким приоритетом занимают много времени процессора, то процессы с более низким приоритетом могут голодать и будут отложены на неопределенное время.~~
* ~~Этот алгоритм планирования может оставить некоторые процессы с низким приоритетом ожидающими бесконечно долго.~~
* ~~Процесс будет заблокирован, когда он будет готов к запуску, но должен ждать ЦП, потому что в данный момент выполняется какой-то другой процесс.~~
* ~~Если новый процесс с более высоким приоритетом продолжает поступать в очередь готовности, то процессу, который находится в состоянии ожидания, может потребоваться длительное время ожидания.~~
* 
* Процессы создаются (пользователем) динамически через форму создания процесса
* В основе архитектуры программы лежит паттерн MVC (model view controller)
* view программы разделена на 4 области:

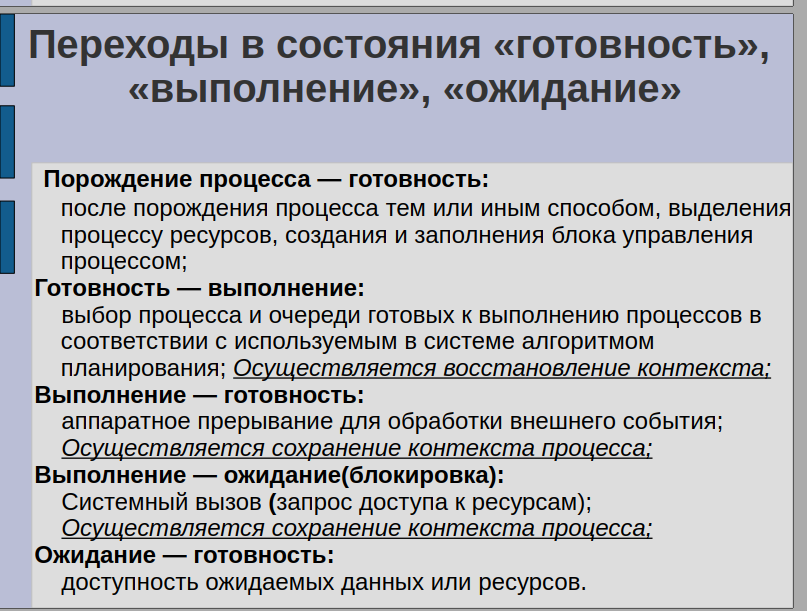
1. список процессов с его характеристиками (название, время выполнения, время прибытия, приоритет)



1. область для вывода сообщений о происходящих действиях в программе
2. область со списками очередей процессов с разными приоритетами



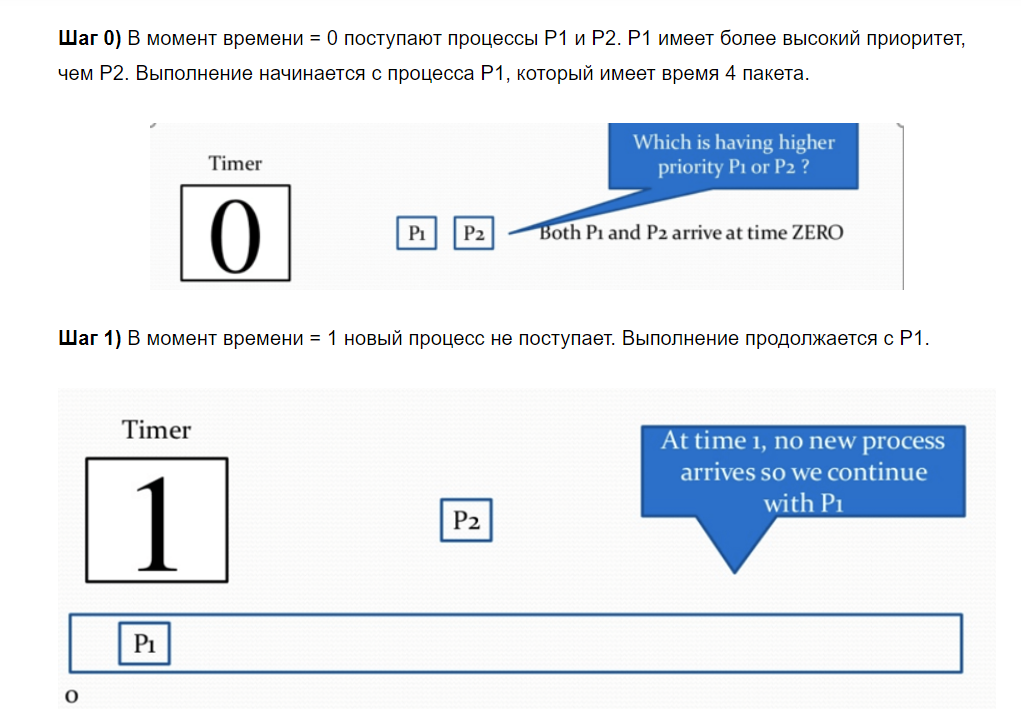
* У каждого процесса есть контекст (состояние), ресурсы. Возможные состояния: готовность, выполнение, ожидание. При изменении состояния из состояния ВЫПОЛНЕНИЕ, происходит сохранение контекста (если это не удаление процесса).



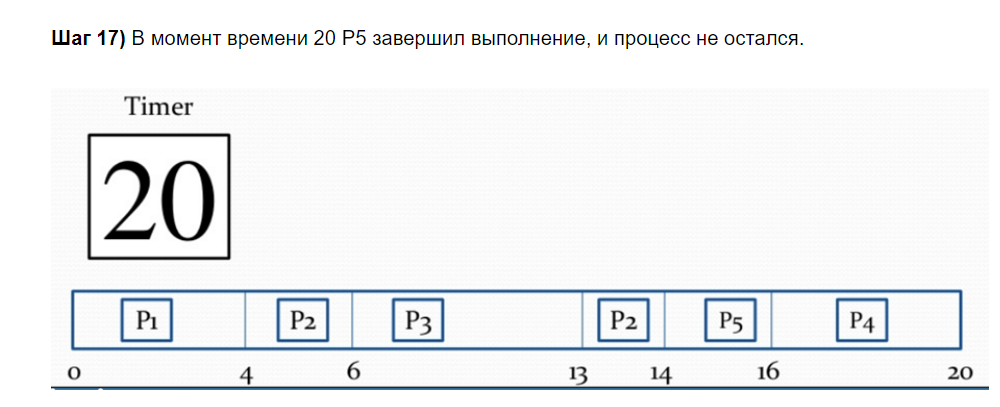
* Алгоритм «поощрений и штрафов» заключается в том, что:

когда задача с более низким приоритетом выполняется и в момент её выполнения приходит другая задача с более высоким приоритетом, задача с низким приоритетом отправляется в ожидание (в очередь) с повышенным приоритетом. То, насколько он будет повышен зависит от времени, которое процесс успел отработать. Но как зависит? С каждым сигналом таймера понижается приоритет выполняемого процесса? Получается, поощрение — это замена низко приоритетной задачи на более приоритетную в момент выполнения? Поощрение для более приоритетной, штраф — для менее приоритетной.

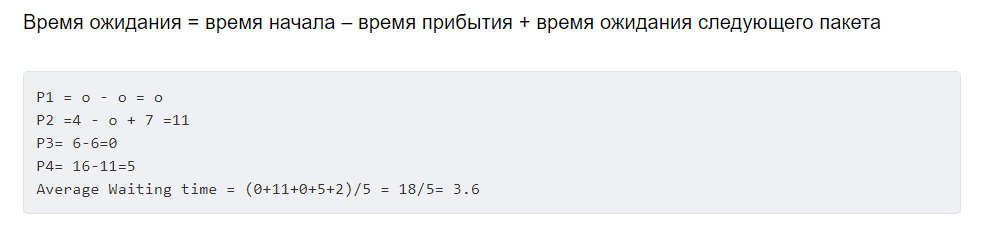
* Как будет отображаться основное, что происходит в программе?



………………



* Подсчет времени ожидания



**Литература/ссылки/…**

про систему динамического планирования

https://coderlessons.com/tutorials/akademicheskii/osnovy-operatsionnykh-sistem/25-algoritm-prioritetnogo-planirovaniia

https://ru.wikipedia.org/

<https://studfile.net/>

<https://coderlessons.com/tutorials/akademicheskii/osnovy-operatsionnykh-sistem/25-algoritm-prioritetnogo-planirovaniia>

https://russianblogs.com/article/8572991004/