

Разработка механизма планирования деятельности процессов в операционной системе.

1. Планирование в системах пакетной обработки данных (batch)

Некоторые алгоритмы используются как в системах планирования данных в блоке, так и в интерактивных системах.

Алгоритм «первым пришел, первым обслужен» — самый простой из алгоритмов планирования. Процессы, обращающиеся к процессору, обслуживаются в порядке поступления запросов. Чаще всего формируется одна очередь ожидающих процессов. Как только появляется первый процесс, он сразу запускается и работает столько, сколько необходимо. Остальные процессы помещаются в конец очереди. Когда текущий процесс блокируется, запускается следующий в очереди, а когда он освобождается от блокировки, процесс попадает в конец очереди. Главным преимуществом этого алгоритма является то, что он прост для понимания и так же прост для программирования. В этом алгоритме все процессы, находящиеся в состоянии ожидания, управляются цепочкой списков. Чтобы выбрать процесс для запуска, нужно просто взять первый элемент из списка и затем удалить его. Появление нового процесса приводит к тому, что он помещается в конец списка.

Алгоритм «самая короткая задача первая» — это алгоритм, который выбирает из списка ожидания процесс с самым коротким временем выполнения. Этот алгоритм предполагает, что периоды времени выполнения процессов известны заранее. Если в очереди есть несколько одинаково важных процессов, планировщик сначала выбирает самую короткую задачу.

Алгоритм наименьшего оставшегося времени выполнения — это алгоритм, который каждый раз выбирает процесс с наименьшим оставшимся временем выполнения. В этом случае необходимо заранее знать время выполнения процесса. Когда поступает новый процесс, его общее время выполнения сравнивается с оставшимся временем выполнения активного процесса. Если время выполнения нового процесса короче, текущий процесс приостанавливается и управление передается новому процессу. Эта схема позволяет быстро обслуживать короткие запросы.

2. Планирование в интерактивных системах

Алгоритм планирования по циклу (round-robin). Каждому процессу предоставляется определенный интервал времени процессора, так называемый квант времени. Если процесс все еще выполняется в конце кванта времени, он завершается, и управление передается следующему процессу. Если процесс зависает или завершается раньше, в этот момент происходит переход управления. Реализация алгоритма round-robin проста. Планировщик должен только поддерживать список процессов в состоянии готовности в виде цикла. Когда процесс достиг своего временного лимита, он отправляется в конец списка, переходя указатель на следующий процесс.

Алгоритм с приоритетным планированием. Основная идея проста: каждому процессу присваивается приоритет, и управление передается процессу с наибольшим приоритетом. Даже на персональном компьютере с одним пользователем может происходить несколько процессов, некоторые из которых более важны, чем другие. Демон, отвечающий за отправку электронной почты в фоновом режиме, имеет более низкий приоритет, чем процесс отображения видео в реальном времени на экране.

Чтобы предотвратить неограниченное выполнение процессов с высоким приоритетом, планировщик может понижать приоритет процесса с каждым тактом (то есть с каждым прерыванием таймера). Если в результате приоритет текущего процесса будет ниже приоритета следующего процесса, произойдет переключение. Каждому процессу можно присвоить максимальное время выполнения. Как только время истекло, управление передается следующему приоритетному процессу. Приоритеты процесса могут назначаться статически или динамически.

Алгоритм с несколькими рядами (классами) — это алгоритм с приоритетными классами. Процессам из класса с наивысшим приоритетом выделено одно квант, процессам из следующего класса выделено два кванта, а процессам из следующего класса выделено три кванта и т. д. Когда процесс использовал все выделенное ему время, он перемещается на класс ниже.

Алгоритм «самый короткий процесс» — это алгоритм, похожий на алгоритм «самая короткая задача» в системах блочной обработки данных.

Алгоритм с честным планированием — это алгоритм, при котором система уделяет внимание владельцу процесса перед планированием. В этой модели каждый пользователь получает часть процессора, и планировщик выбирает процесс в соответствии с этим. Если каждому пользователю было обещано 50% процессора, то он получит 50% процессора, независимо от количества процессов пользователей.

3. Практическая часть

Для выполнения заданы 4 процесса A, B, C, D со значениями времени отклика (время выполнения) 4, 20, 12, 8 единиц.

3.1. Алгоритм «первым пришел, первым выполнен»

Порядок выполнения A B C D

Средняя продолжительность составляет 11 единиц. $(4+20+12+8)/4 = 11$

Время выполнения для каждого процесса:

первый процесс — за 4 единицы,

второй — за $4+20=24$ единицы,

третий - за $4+20+12= 36$ единиц,

четвертый — за $4+20+12+8= 44$ единицы.

Среднее время выполнения будет равно $4a + 3b + 2c + d)/4$.

$(4*4+3*20+2*12+8)/4=27$ единиц

3.2. Алгоритм кратчайшего задания является первым

Порядок выполнения A D C B

Средняя продолжительность составляет 11 единиц. $(4+8+12+20)/4 = 11$

Время выполнения каждого процесса:

первый процесс — за 4 единицы,

второй - за $4+8=12$ единиц,

третий — за $4+8+12= 24$ единицы,

четвертый — за $4+8+12+20= 44$ единицы.

Среднее время выполнения будет равно $4a + 3b + 2c + d)/4$.

$(4*4+3*8+2*12+20)/4=21$ единиц

3.3. Алгоритм кратчайшего оставшегося времени выполнения

Может быть выполнен только за время. $K= 10$.

Порядок выполнения A D C(2) C B(10) B

Средняя продолжительность составляет 11 единиц. $(4+8+12+20)/4 = 11$

Время выполнения для каждого процесса:

первый процесс — за 4 единицы,

второй - за $4+8=12$ единиц,

третий – в $4+8+12= 24$ единицы,

четвертый – в $4+8+12+20= 44$ единиц.

Среднее время пробега будет равно $6a + 5b + 4c + 3d+2e+f)/6$.

3.4. Алгоритм планирования в цикле (round-robin).

Временной интервал составляет $K=13$

Порядок выполнения A B(7) C D B

Среднее время выполнения будет равно $5b + 4c + 3d + 2e + f)/5$.

$$(5*4 + 4*13 + 3*12 + 2*8 + 7)/5 = 26,2$$

3.5. Алгоритм с приоритетным планированием.

Приоритеты процессов будут A-3 B-8 C-5 D-4

Порядок выполнения B C D A

Среднее время выполнения будет равно $4a + 3b + 2c + d)/4$.

$$(4*20 + 3*12 + 2*8 + 4)/4 = 24$$

3.6. Алгоритм кратчайшего процесса следующий

Количество времени составляет $K=8$

Порядок выполнения A D C(4) C B(12) B(4) B

Среднее время выполнения будет равно $7a + 6b + 5c + 4d + 3e + 2f + g)/7$.

$$(7*4 + 6*8 + 5*8 + 4*4 + 3*8 + 2*8 + 4)/7 = 25,14$$

3.7. Алгоритм с честным планированием

Задача: При выполнении заданы процессы трех пользователей.

Первый пользователь получает 50% времени процессора. Он передает на выполнение 3 процесса:

A с временем выполнения 10 единиц, B с временем выполнения 12 единиц, C с временем выполнения 7 единиц.

Второй пользователь получает 25% времени процессора. Он передает на выполнение 2 процесса:

X с временем выполнения 6 единиц, Y с временем выполнения 10 единиц.

Третий пользователь получает 25% времени процессора. Он передает на выполнение 1 процесс:

Z с временем выполнения 26 единиц.

Временной квант составляет $K=8$ единиц.

Какой будет порядок выполнения процессов для алгоритма «первым пришел, первым выполнен»? Ответ: A(2)B(4)XZ(18)CAY(2)Z(10)BYZ(2)Z

Какой будет порядок выполнения процессов для алгоритма «первым пришел, первым выполнен», если все три пользователя получают по 33% времени пользователя?

Ответ: A(2)XZ(18)B(4)Y(2)Z(10)CYZ(2)AZB

Лабораторная работа № 4

Тема работы: Разработка механизма планирования деятельности процессов в операционной системе.

Основная цель: Изучить и разработать механизм планирования процессов в операционной системе, понимая алгоритмы планирования и их практическую реализацию для оптимизации использования ресурсов и времени выполнения.

Конкретные задачи:

- Ознакомление с основными концепциями процессов и их состояний в операционных системах.
- Изучение роли планировщика (scheduler) и типов планирования: для интерактивных систем, систем пакетной обработки данных (batch).
- Анализ классических алгоритмов планирования процессов: **FCFS (First Come First Served)**, **SJF (Shortest Job First)**, **Round Robin**, **Приоритеты** и т.д.
- Практическая реализация (путем моделирования) некоторых алгоритмов планирования процессов.
- Сравнение производительности алгоритмов (время ожидания, время отклика, время оборота).
- Формулировка выводов об эффективности различных методов планирования и их применимости в реальных системах.

Практические задачи:

1. Алгоритм «Первым пришел – первым обслужен» (First Come, First Served – FCFS)

1. ****Входные данные**:** A(5), B(9), C(4), D(7). Найдите среднее время выполнения и время для каждого процесса.
2. ****Входные данные**:** P1(6), P2(3), P3(8), P4(2). Найдите среднее время выполнения и время для каждого процесса.
3. ****Входные данные**:** X(2), Y(6), Z(3), W(4). Найдите среднее время выполнения и время для каждого процесса.
4. ****Входные данные**:** T1(10), T2(2), T3(4), T4(6). Найдите среднее время выполнения и время для каждого процесса.
5. ****Входные данные**:** M(7), N(5), O(8), P(3). Найдите среднее время выполнения и время для каждого процесса.

2. Алгоритм «Самый короткий процесс первым» (Shortest Job First – SJF)

1. ****Входные данные**:** A(7), B(2), C(4), D(1). Найдите среднее время выполнения и время для каждого процесса.
2. ****Входные данные**:** P1(5), P2(3), P3(1), P4(2). Найдите среднее время выполнения и время для каждого процесса.
3. ****Входные данные**:** X(6), Y(4), Z(3), W(2). Найдите среднее время выполнения и время для каждого процесса.
4. ****Входные данные**:** T1(8), T2(3), T3(6), T4(1). Найдите среднее время выполнения и время для каждого процесса.
5. ****Входные данные**:** M(9), N(5), O(2), P(4). Найдите среднее время выполнения и время для каждого процесса.

3. Алгоритм «Самое короткое оставшееся время» (Shortest Remaining Time First – SRTF)

1. ****Входные данные****: A(8), B(4), C(3), D(7). K = 5. Найдите среднее время выполнения и порядок выполнения процессов.
2. ****Входные данные****: P1(10), P2(5), P3(6), P4(8). K = 4. Найдите среднее время выполнения и порядок выполнения процессов.
3. ****Входные данные****: X(7), Y(9), Z(5), W(3). K = 3. Найдите среднее время выполнения и порядок выполнения процессов.
4. ****Входные данные****: T1(11), T2(6), T3(4), T4(8). K = 6. Найдите среднее время выполнения и порядок выполнения процессов.
5. ****Входные данные****: M(9), N(2), O(8), P(5). K = 4. Найдите среднее время выполнения и порядок выполнения процессов.

4. Алгоритм «Round Robin» (Round Robin – RR)

1. ****Входные данные****: A(10), B(15), C(20), D(25). K = 10. Найдите среднее время выполнения и порядок выполнения процессов.
2. ****Входные данные****: P1(12), P2(8), P3(16), P4(6). K = 7. Найдите среднее время выполнения и порядок выполнения процессов.
3. ****Входные данные****: X(9), Y(14), Z(11), W(13). K = 5. Найдите среднее время выполнения и порядок выполнения процессов.
4. ****Входные данные****: T1(20), T2(12), T3(9), T4(16). K = 8. Найдите среднее время выполнения и порядок выполнения процессов.
5. ****Входные данные****: M(18), N(22), O(13), P(17). K = 9. Найдите среднее время выполнения и порядок выполнения процессов.

5. Алгоритм «Планирование на основе приоритетов» (Priority Scheduling)

1. ****Входные данные****: A(5, 1), B(9, 3), C(6, 2), D(8, 4). Найдите среднее время выполнения и порядок выполнения процессов.
2. ****Входные данные****: P1(10, 2), P2(4, 1), P3(8, 3), P4(6, 4). Найдите среднее время выполнения и порядок выполнения процессов.
3. ****Входные данные****: X(12, 2), Y(7, 1), Z(5, 3), W(9, 4). Найдите среднее время выполнения и порядок выполнения процессов.
4. ****Входные данные****: T1(11, 3), T2(6, 1), T3(9, 2), T4(8, 4). Найдите среднее время выполнения и порядок выполнения процессов.
5. ****Входные данные****: M(13, 4), N(10, 3), O(7, 2), P(5, 1). Найдите среднее время выполнения и порядок выполнения процессов.

6. Задачи для алгоритма «Самый короткий процесс следующий» (Временной квант K=8)

1. ****Входные данные****: 4 процесса со временем выполнения: A = 5, B = 12, C = 7, D = 9. Найдите порядок выполнения и среднее время выполнения.
2. ****Входные данные****: 5 процессов со временем выполнения: A = 15, B = 3, C = 8, D = 20, E = 4. Найдите порядок выполнения и среднее время выполнения.
3. ****Входные данные****: 4 процесса со временем выполнения: A = 10, B = 8, C = 6, D = 2. Найдите порядок выполнения и среднее время выполнения.
4. ****Входные данные****: 3 процесса со временем выполнения: A = 6, B = 16, C = 8. Найдите порядок выполнения и среднее время выполнения.
5. ****Входные данные****: 4 процесса со временем выполнения: A = 3, B = 9, C = 14, D = 6. Найдите порядок выполнения и среднее время выполнения.

7. Задачи для алгоритма «Планирование на основе справедливости» (Временной квант K=8)

1. ****Входные данные****: три пользователя. Первый пользователь получает 33% времени процессора и имеет процессы A(12), B(8). Второй пользователь получает 33% времени и имеет процессы X(10), Y(6). Третий пользователь получает 33% времени и имеет процесс Z(16). Какой будет порядок выполнения процессов?
2. ****Входные данные****: Три пользователя. Первый пользователь получает 50% времени процессора и имеет процессы A(10), B(5). Второй пользователь получает 25% времени и имеет процессы X(7), Y(3). Третий пользователь получает 25% времени и имеет процесс Z(12). Какой будет порядок выполнения процессов?
3. ****Входные данные****: три пользователя. Первый пользователь получает 33% времени процессора и имеет процессы A(8), B(6), C(5). Второй пользователь получает 33% времени и имеет процессы X(12), Y(10). Третий пользователь получает 33% времени и имеет процесс Z(20). Какой будет порядок выполнения процессов?
4. ****Входные данные****: Четыре пользователя. Первый пользователь получает 40% времени процессора и имеет процессы A(15), B(10), C(8). Второй пользователь получает 20% времени и имеет процессы X(6), Y(5). Третий пользователь получает 20% времени и имеет процесс Z(18). Четвертый пользователь получает 20% времени и имеет процесс Q(14). Какой будет порядок выполнения процессов?
5. ****Входные данные****: Три пользователя. Первый пользователь получает 50% времени процессора и имеет процессы A(10), B(7), C(6). Второй пользователь получает 25% времени и имеет процесс X(10). Третий пользователь получает 25% времени и имеет процесс Z(14). Какой будет порядок выполнения процессов?

Критерии оценки:

Для получения оценки 5-6 обязательны задания 1, 5, 6, 7.

Для получения оценки 7-8 обязательны задания 1, 2, 3, 5, 6, 7.

Для получения оценки 9-10 обязательны задания 1, 2, 3, 4, 5, 6, 7.

Если использовался ИИ (искусственный интеллект), оценка снижается на 2 балла, только если студент дал объяснения в функциональной части. В противном случае лабораторная работа не принимается.

Контрольные вопросы:

1. Что такое **процесс** и какие состояния он может иметь в операционной системе?
2. Какова роль **планировщика процессов (scheduler)** и какие типы планирования существуют (долгосрочное, среднесрочное, краткосрочное)?
3. В чем разница между **превентивным планированием** и **непревентивным планированием**?
4. Объясните принцип работы классических алгоритмов планирования:
5. **FCFS (First Come, First Served)**
6. **SJF (Shortest Job First)**
7. **Round Robin**
8. **Планирование на основе приоритетов.**
9. Какие показатели производительности используются для оценки алгоритмов планирования (время ожидания, время отклика, время оборота, степень использования процессора)?
10. Каковы преимущества и недостатки алгоритма Round Robin по сравнению с FCFS и SJF?
11. Как **размер кванта времени** влияет на производительность в алгоритме Round Robin?
12. В каких практических ситуациях предпочтительнее использовать простой алгоритм (FCFS), а в каких необходимы более сложные алгоритмы (SJF, приоритеты, Round Robin)?
13. Как можно моделировать алгоритмы планирования процессов (например, с помощью таблиц, диаграмм Ганта или программ)?

14. Почему механизм планирования процессов важен для **справедливости** и **эффективности** использования системных ресурсов?

Список рекомендуемой литературы:

1. С. Ғапуҗ, А. Ғапуҗ – *Операционные системы. Основные принципы и механизмы*, издательство Politehnica Press, Бухарест.
2. Г. Текучи, Д. Кэлушер – *Операционные системы и их администрирование*, издательство Matrix Rom, Бухарест.
3. М. Вочин – *Вычислительные структуры и операционные системы*, Издательство Университета Бухареста.
4. Материалы курса и лабораторных работ факультетов информатики и автоматике Румынии.
5. А. С. Таненбаум, Х. Бос – *Современные операционные системы*, 4-е издание, СПб: Питер, 2017.
6. В. М. Бурков, А. В. Духовный – *Операционные системы: теория и практика*, Москва: Академия, 2019.
7. Е. В. Додонов – *Операционные системы и среды*, Донецк: ДонНТУ, 2016.
8. Учебные пособия по моделированию и планированию процессов в ОС (доступные в университетских репозиториях РФ и СНГ).
9. Абрахам Сильбершац, Питер Бэр Галвин, Грег Гэнь – *Концепции операционных систем*, 10-е издание, Wiley, 2018.
10. Эндрю С. Таненбаум, Герберт Бос – *Современные операционные системы*, 4-е издание, Pearson, 2015.
11. William Stallings – *Operating Systems: Internals and Design Principles*, 9th Edition, Pearson, 2018.
12. Ремзи Х. Арпачи-Дюссо, Андреа К. Арпачи-Дюссо – *Операционные системы: три простых части*, Agraci-Dusseau Books, 2018 (доступно бесплатно онлайн: ostep.org).