

К. А. Кондратьев, Н. Н. Шумаков, А. А. Колташев

## АЛГОРИТМЫ ДИСПЕТЧЕРИЗАЦИИ ВЫЧИСЛЕНИЙ И ИХ ПРИМЕНЕНИЕ В БОРТОВОМ ПРОГРАММНОМ ОБЕСПЕЧЕНИИ СПУТНИКОВ СВЯЗИ И НАВИГАЦИИ

*Рассматриваются основные виды алгоритмов диспетчеризации, их особенности, классификация и применение при построении мультипрограммных вычислительных систем, а также реализация их в бортовом программном обеспечении спутников связи и навигации.*

*Ключевые слова:* алгоритмы диспетчеризации, дисциплины обслуживания, бортовое программное обеспечение.

Одной из основных задач, выполняемых любой операционной системой, является задача распределения (планирования) ресурсов вычислителей между пользователями или выполняемыми вычислительными процессами (задачами). Основным ресурсом, требующим организованного распределения, является процессор. При построении ядра управляющей вычислительным процессом программы (диспетчера) может использоваться целый ряд различных методов и алгоритмов диспетчеризации вычислений.

Анализируя существующие методы организаций вычислений, можно выделить несколько основных принципов, дисциплин и аспектов реализации алгоритмов диспетчеризации:

- вытесняющие и невытесняющие дисциплины алгоритмов диспетчеризации;
- диспетчеризация на основе времени готовности к обслуживанию;
- диспетчеризация на основе приоритетов;
- диспетчеризация с квантованием;
- диспетчеризация с учетом хода выполнения процесса и его характеристик;
- диспетчеризация с применением механизмов межзадачного взаимодействия.

*Вытесняющие и невытесняющие дисциплины алгоритмов диспетчеризации.* Невытесняющая многозадачная дисциплина обслуживания – это такой способ организации обслуживания процессов, при котором текущий активный процесс выполняется до тех пор, пока сам и по своей инициативе не отдаст управление диспетчеру для выбора и запуска следующего готового к исполнению процесса.

Вытесняющая многозадачная дисциплина обслуживания – это такой способ организации обслуживания, при котором решение о переключении к исполнению другого готового процесса принимает диспетчер, а не текущий активный процесс. При этом операционная система решает ряд следующих функций:

- определение момента переключения процессов;
- сохранение контекста текущей задачи;
- выбор на выполнение следующей готовой задачи;
- загрузка контекста выбранной задачи и запуск ее на исполнение.

*Диспетчеризация на основе времени готовности к обслуживанию.* Самым распространенным параметром, лежащим в основе диспетчеризации, является время перехода в состояние готовности к обслуживанию вычис-

лительного процесса. Анализируя существующие алгоритмы диспетчеризации, мы можем разделить их на следующие три группы:

- 1) диспетчеризация по принципу очереди: первым пришел – первым обслужен;
- 2) диспетчеризация по принципу стека: последним пришел – первым обслужен;
- 3) круговая диспетчеризация: последовательное выполнение процессов вне зависимости от их времени начала готовности к выполнению.

*Диспетчеризация на основе приоритетов.* При организации приоритетного распределения времени выполнения процессов, диспетчеризацию можно разделить по типу применяемых приоритетов:

- статические – вычислительные процессы имеют фиксированное значение приоритета. Как правило, распределение приоритетов проводится на этапе проектирования системы;
- динамические изменяемые – приоритеты вычислительных процессов могут меняться при необходимости и в зависимости от хода вычислений.

Приоритетную диспетчеризацию также можно разделить по объекту присвоения приоритета:

- диспетчеризация на основе приоритетов вычислительных процессов. Каждому процессу (задаче) сопоставляется определенное значение приоритета, в соответствии с которым он получает требуемый ему ресурс;
- диспетчеризация на основе приоритетов групп задач. Задачи делят на группы, которым присваивается определенное значение приоритета (в рамках своей группы задачи равноправны);
- диспетчеризация на основе приоритетов очередей.

Ряд операционных систем реализует приоритетную диспетчеризацию путем сопоставления приоритетов не процессам и группам из них, а очередям готовых к выполнению задач. Каждой очереди соответствует свой уровень приоритета. Задача, приоритет которой следует изменить, перемещается в очередь соответствующего приоритета. В качестве примера можно привести алгоритмы диспетчеризации, использующие временные кванты: задача, выполнение которой превысило выделенный ей квант времени, переводится в очередь готовых к выполнению задач с более низким приоритетом и продолжит свое выполнение в случае отсутствия задач в очередях с более высоким значением приоритета.

*Диспетчеризация с квантованием.* В ряде операционных систем в процесс распределения ресурса между

потребителями вводится понятия кванта. Квант – это максимальный выделенный промежуток времени (порция), в течение которого ресурс предоставляется одному пользователю. Применяя это понятие к диспетчеризации процессорного времени, мы можем сказать, что каждый процесс выполняется на строго ограниченном промежутке времени (кванте), после чего управление передается следующему готовому к выполнению процессу.

Алгоритмы диспетчеризации с квантованием можно разделить на два вида:

- алгоритмы с равными квантами – каждому процессу в равной степени предоставляются одинаковые интервалы исполнения;

- алгоритмы с разной величиной квантов – процессам выделяются кванты различной длины в зависимости от условий, приоритетности и иных характеристик процесса.

*Диспетчеризация с учетом хода выполнения процесса.* Во многих операционных системах, в целях оптимизации хода вычислений, вводят систему штрафов по отношению к конкурирующим за ресурс процессам. Диспетчер уменьшает приоритет выполняющегося процесса в случае его некорректной работы. Некорректностью может быть превышение заявленного времени выполнения задачи, программные ошибки, содержащиеся в ней, а также любые запрещенные данному процессу действия, такие как, например, попытка записи в чужую область памяти, нарушение доступа к данным и т. д.

*Диспетчеризация с учетом характеристик выполняющегося процесса.* В ряде операционных систем, при организации распределения процессорного времени посредством сопоставления приоритетов процессам, учитываются их характеристики. Это может быть тип пользователя, от имени которого запущен процесс, требуемые процессом ресурсы (объем памяти, количество устройств ввода-вывода, с которым он работает), а также иные специфические характеристики процесса. Учет характеристик процесса при диспетчеризации позволяет оптимизировать общий ход вычислений.

Диспетчеризация с применением механизмов межзадачного взаимодействия.

Диспетчер любой мультипрограммной системы осуществляет координацию процессов и планирование очередности выполнения задач. Диспетчер получает управление при возникновении любого события, требующего прерывания выполнения текущего процесса. Это может быть внешнее прерывание от аппаратуры либо появление заявки на выполнения другого более приоритетного вычислительного процесса. В тех случаях, когда несколько процессов являются взаимозависимыми, и появляется необходимость обеспечения их синхронизации, в систему вводят ряд механизмов синхронизации межзадачного взаимодействия, самым распространенным из которых является семафор. Он представляет собой объект, содержащий численное значение. Выполняющиеся процессы могут осуществлять над ним функции захвата и освобождения. При захвате семафора его значение уменьшается на единицу, при освобождении – увеличивается; в случае нулевого значения семафора выполняющий операцию захвата процесс переходит в режим ожидания.

Бинарный (двоичный) вариант семафора называется мьютексом, он широко применяется при синхронизации межзадачного взаимодействия в случае необходимости обеспечения доступа к общему ресурсу, владеть которым в определенный момент может только один потребитель. Общим ресурсом может быть файл, информационное поле или внешнее устройство ввода-вывода. Достаточно часто применяется вариант семафора с приоритетом, процесс, захвативший семафор, применяет к себе его приоритет, как правило, имеющий большее значение.

Состояния процессов при организации мультипрограммных систем. При организации мультипрограммной системы с диспетчеризацией вычислительных процессов, вводится понятие текущего состояния процесса. В различных операционных системах вычислительные процессы могут быть в следующих состояниях:

- выполнение – процесс (задача) в данный момент осуществляется процессором;

- готовность к выполнению – новый процесс не выполняется и ожидает возможности своего выполнения;

- прерванное – действие процесса было приостановлено для передачи управления более приоритетному процессу, процесс находится в состоянии ожидания возможности продолжить свое выполнение;

- ожидание – процесс находится в прерванном состоянии до выполнения определенного набора заданных условий;

- приостановка – процесс не является подходящим для выполнения. Данное состояние используется в особых случаях.

Часть вычислительных систем, в зависимости от своего целевого назначения, позволяет прерывать выполнение не только пользовательских задач, но и выполнение самого ядра системы. Таким образом, достигается максимально быстрая реакция системы на внешние события. В свою очередь, при критичности выполнения текущего процесса, обработка прерываний может сводиться только к сохранению информации о нем (фиксация факта самого события), непосредственно обработка прерывания ядром системы будет произведена позже.

*Диспетчеризация вычислительного процесса в бортовом программном обеспечении спутников связи и навигации.* В бортовом программном обеспечении (БПО) спутников связи и навигации реализован мультипрограммный режим работы с приоритетной дисциплиной обслуживания задач с вытесняющей многозадачностью. Приоритеты являются постоянными и абсолютными, назначаются на этапе проектирования, исходя из общесистемных интересов, с учетом допустимого времени обслуживания задачи и времени ее работы. Каждой задаче соответствует определенное значение приоритета.

Приоритет задачи – число [0..255]. Наиболее приоритетная задача – задача с наименьшим численным значением приоритета. Первые несколько значений приоритетов зарезервированы для системных задач, а последнее значение 255 используется для специальной задачи «фоновая», которая выполняется процессором в случае отсутствия в системе любых других готовых к выполнению задач. Приоритет 0 является признаком запуска задачи с запретом внешних прерываний.

При совпадении приоритетов более приоритетной является прерванная (ранее выполнявшаяся задача). При совпадении приоритетов прерванных задач, более приоритетной является та, которая поступила в очередь готовых задач раньше. При совпадении приоритетов для новых задач, более приоритетной задачей считается задача с меньшим номером. При совпадении приоритетов новой и прерванной задачи, более приоритетной считается прерванная задача.

Операционная система предоставляет ряд специальных функций для изменения приоритетов задач во время работы. Приоритеты могут быть изменены статически – путем корректировки таблицы задач, после чего задачи будут запускаться с новыми значениями приоритетов, либо динамически – изменение значения приоритета во время выполнения задачи.

В целях синхронизации межзадачного взаимодействия операционная система предоставляет ряд механизмов: двоичные семафоры с приоритетами, почтовые сообщения, флаги синхронизации, функции задержки выполнения задачи и запрет прерываний.

Соответствие приоритетов задачам и семафорам определяется на этапе проектирования. При некорректном распределении приоритетов между задачами, могут возникнуть «клинчи» вычислительного процесса.

Примером может случиться конкуренция двух задач по захвату ресурсов. Две задачи захватили по одному ресурсу, входят в состояния ожидания освобождения, необходимого для продолжения работы ресурса, захваченного параллельной задачей. Таким образом, обе задачи находятся в прерванном состоянии, захватив ресур-

сы, необходимые друг другу для продолжения работы. Ситуация может усложниться многомерностью «клинча», когда целая цепочка вычислительных процессов таким образом блокирует выполнение друг друга.

Для решения данной проблемы ОС реализует протокол предельного приоритета при работе с бинарными (двоичными) семафорами с приоритетами:

– каждому семафору назначается приоритет, который больше или равен приоритету всех задач, использующих этот семафор; меньше приоритета всех задач, не использующих этот семафор, но более приоритетных, чем те, что его используют;

– когда задача получает семафор, приоритет задачи «немедленно» устанавливается к приоритету семафора;

– когда задача отдает семафор, ее приоритет устанавливается назад к приоритету, какой был у нее до захвата семафора.

Таким образом, если в данный момент семафор уже используется какой-либо задачей, другая задача, которой он понадобится, не запустится на выполнение или не продолжится, до его освобождения.

Приоритетная дисциплина обслуживания и мультипрограммный режим выполнения вычислений в настоящее время являются необходимыми условиями реализации любой современной и эффективной операционной системы реального времени. Алгоритмы диспетчеризации с динамически изменяющимися приоритетами и вытесняющей дисциплиной обслуживания позволяют оптимизировать общий ход вычислений, улучшать время реакции системы на события, а также делать дальнейшее сопровождение системы удобным и более эффективным.

C. A. Kondratev, N. N. Shumakov, A. A. Koltashev

## **COMPUTING ALGORITHMS DISPATCHING AND USE IN NAVIGATION AND COMMUNICATION SATELLITES ON-BOARD SOFTWARE**

*The article considers the main types of algorithms of dispatching, their special features, classification and their usage in design of multi-program computing system, as well as their realization in navigation and communication satellites on-board software.*

*Keywords: dispatching algorithms, service procedures, on-board software.*

© Кондратьев К. А., Шумаков Н. Н., Колташев А. А., 2010