

# prionctl(1)

## НАЗВАНИЕ

**prionctl** - выдача или установка параметров планировщика для указанных процессов.

## СИНТАКСИС

```
prionctl -l
prionctl -d [ -i тип_идентификатора ] [ список_идентификаторов ]
prionctl -s [ -c класс ] [ специфические_опции_класса ]
    [ -i тип_идентификатора ] [ список_идентификаторов ]
prionctl -e [ -c класс ] [ специфические_опции_класса ]
команда [ аргументы ]
```

## ОПИСАНИЕ

Команда **prionctl** выдает или устанавливает *параметры планировщика* (scheduling parameters) для указанного процесса или процессов. Ее можно также использовать для выдачи текущей информации о конфигурации планировщика процессов системы или выполнения команды с явно заданными параметрами планировщика.

Процессы разбиваются на разные классы, для каждого из которых применяются свои правила планирования. В настоящее время поддерживаются следующие классы: процессы реального времени, процессы с разделением времени и интерактивные процессы. Характеристики этих классов и специфические опции, которые они поддерживают, описаны ниже в подразделах "**Процессы реального времени**", "**Процессы с разделением времени**" и "**Интерактивные процессы**" раздела "**ИСПОЛЬЗОВАНИЕ**". При наличии соответствующих прав доступа, команда **prionctl** позволяет изменить класс и другие параметры планировщика для работающего процесса.

В стандартной конфигурации готовые к работе процессы реального времени выполняются перед любыми другими процессами. Поэтому неправильное использование процессов реального времени может существенно снизить производительность системы.

Если указывается **список\_идентификаторов**, он должен задаваться в командной строке последним, а элементы этого списка должны разделяться пробелами. Если **список\_идентификаторов** не задан, аргумент **тип\_идентификатора** со значением **pid**, **ppid**, **pgid**, **sid**, **taskid**, **class**, **uid**, **gid** или **projid** задают, соответственно, идентификатор процесса, идентификатор родительского процесса, идентификатор группы процессов, идентификатор сеанса, идентификатор задания, класс, идентификатор пользователя, идентификатор группы или идентификатор проекта самой команды **prionctl**.

Команда:

```
prionctl -d [-i тип_идентификатора] [список_идентификаторов]
```

выдает информацию о классе и специфические параметры планировщика для классов процессов, задаваемых параметрами **тип\_идентификатора** и **список\_идентификаторов**.

Команда:

```
priocntl -s [-c класс] [специфические_опции_класса] [-i тип_идентификатора]  
[список_идентификаторов]
```

устанавливает заданные в командной строке класс и специфические параметры планировщика для указанных процессов. Опция **-c класс** задает класс, к которому надо отнести эти процессы. (Допускаются аргументы: **RT** - процессы реального времени (real-time), **TS** - процессы с разделением времени (time-sharing) и **IA** - интерактивные процессы (inter-active).)

Устанавливаемые специфические параметры планировщика для класса задаются специфическими опциями класса, как описано далее в соответствующем разделе. Если опция **-c класс** не указана, параметры **тип\_идентификатора** и **список\_идентификаторов** должны задавать набор процессов одного класса, иначе будет выдано сообщение об ошибке. Если **специфические\_опции\_класса** не указаны, специфические параметры планировщика для процесса сбрасываются в стандартные значения класса, заданного опцией **-c класс** (или в стандартные значения текущего класса процесса, если опция **-c класс** тоже не указана).

Чтобы изменить параметры планировщика для процесса с помощью команды **priocntl**, реальный и эффективный идентификаторы пользователя и группы у пользователя, вызывающего **priocntl**, должны соответствовать реальному и эффективному идентификатору пользователя и группы указанных процессов, или эффективный идентификатор пользователя должен соответствовать суперпользователю (0). Это минимальные требования для всех классов. Отдельные классы могут налагать дополнительные требования для перевода процессов в соответствующий класс или установки специфических параметров планировщика для класса.

Когда параметры **тип\_идентификатора** и **список\_идентификаторов** задают несколько процессов, порядок обработки этих процессов командой **priocntl** зависит от реализации. Если при обработке одного или нескольких процессов командой **priocntl** возникает ошибка, продолжение дальнейшей обработки зависит от причин, вызвавших эту ошибку.

Если ошибка связана с правами доступа, команда **priocntl** выдает сообщение об ошибке и продолжает обрабатывать процессы в заданном наборе, устанавливая параметры для всех процессов, для которых пользователь имеет необходимые права доступа. Если ошибка **priocntl** вызвана не отсутствием прав доступа, а другими причинами, выдается сообщение об ошибке и обработка набора процессов немедленно прекращается.

Специальный класс процессов **sys** используется для планирования выполнения определенных специальных системных процессов (таких как процесс подкачки страниц, swapper). Класс **sys** нельзя явно установить никакому процессу. Кроме того, любые процессы класса **sys**, входящие в набор процессов, задаваемых параметрами **тип\_идентификатора** и **список\_идентификаторов**, пропускаются командой **priocntl**. Например, если задан **тип\_идентификатора uid**, **список\_идентификаторов** со значением 0 задает все процессы с идентификатором пользователя 0, кроме процессов, относящихся к классу **sys**, и (при изменении параметров с помощью опции **-s**) процесса **init**.

Процесс **init** (с идентификатором процесса 1) обрабатывается специальным образом. Чтобы с помощью команды **priocntl** изменить класс и другие параметры планировщика для процесса **init**, надо задать **тип\_идентификатора pid** и **список\_идентификаторов** с единственным значением 1. Процесс **init** можно отнести к любому классу, сконфигурированному в системе, но класс процессов с разделением времени подходит для него в большинстве случаев. (Другие классы могут оказаться крайне нежелательными; подробнее об этом см. в руководстве по системному администрированию, System Administration Guide, Volume 1.)

Команда:

```
priocntl -e [-c класс] [специфические_опции_класса] команда [аргумент...]
```

выполняет указанную команду с заданными в командной строке классом и параметрами планировщика (при этом ей передаются указанные **аргументы**). Если опция **-c класс** не указана, команда выполняется в текущем классе пользователя.

## ОПЦИИ

Поддерживаются следующие опции:

### **-c класс**

Задаёт класс, который надо установить. (Допускаются аргументы: **RT** - процессы реального времени (real-time), **TS** - процессы с разделением времени (time-sharing) и **IA** - интерактивные процессы (inter-active).) Если указанный класс ещё не сконфигурирован, он конфигурируется автоматически.

### **-d**

Выдаёт параметры планировщика для заданного набора процессов.

### **-e**

Выполняет указанную команду с заданным классом и параметрами планировщика для всех процессов.

### **-i тип\_идентификатора**

Эта опция вместе с аргументами **списка\_идентификаторов** (если они заданы), задаёт один или несколько процессов, к которым необходимо применить команду **priocntl**. Интерпретация **списка\_идентификаторов** зависит от значения **типа\_идентификатора**. Допустимые значения аргумента **тип\_идентификатора** и соответствующие интерпретации **списка\_идентификаторов** представлены ниже:

#### **-i pid**

**Список\_идентификаторов** задаёт идентификаторы процессов. Команда **priocntl** применяется к процессам с перечисленными идентификаторами.

#### **-i ppid**

**Список\_идентификаторов** задаёт идентификаторы родительских процессов. Команда **priocntl** применяется ко всем процессам, идентификаторы родительских процессов которых входят в этот список.

#### **-i pgid**

**Список\_идентификаторов** задаёт идентификаторы групп процессов. Команда **priocntl** применяется ко всем процессам, входящим в указанные группы.

#### **-i sid**

**Список\_идентификаторов** задаёт идентификаторы сеансов. Команда **priocntl** применяется ко всем процессам в указанных сеансах.

#### **-i taskid**

**Список\_идентификаторов** задаёт идентификаторы заданий. Команда **priocntl** применяется ко всем процессам, относящимся к указанным заданиям.

#### **-i class**

**Список\_идентификаторов** состоит из одного имени класса (**RT** - процессы реального времени, **TS** - процессы с разделением времени или **IA** - интерактивные процессы). Команда **priocntl** применяется ко всем процессам указанного класса.

#### **-i uid**

**Список\_идентификаторов** задаёт идентификаторы пользователей. Команда **priocntl** применяется ко всем процессам, эффективный идентификатор пользователя которых совпадает с одним из идентификаторов в списке.

### **-i gid**

**Список\_идентификаторов** задает идентификаторы групп пользователей. Команда **priocntl** применяется ко всем процессам, эффективный идентификатор группы пользователей которых совпадает с одним из идентификаторов в списке.

### **-i projid**

**Список\_идентификаторов** задает идентификаторы проектов. Команда **priocntl** применяется ко всем процессам, эффективный идентификатор проекта которых совпадает с одним из идентификаторов в списке.

### **-i all**

Команда **priocntl** применяется ко всем существующим процессам. **Список\_идентификаторов** задавать не надо (если он задан, то игнорируется). Описанные ниже требования к правам доступа, тем не менее, должны быть выполнены.

Если опция **-i тип\_идентификатора** не указана при использовании опций **-d** или **-s**, предполагается стандартный тип идентификатора - **pid**.

### **-l**

Выдает список классов, сконфигурированный в настоящий момент в системе, вместе со специфической информацией о каждом классе. Формат выдачи специфической информации для класса описан далее в разделе ["ИСПОЛЬЗОВАНИЕ"](#).

### **-s**

Устанавливает параметры планировщика для заданного набора процессов.

При установке параметров для процессов реального времени допустимы следующие специфические опции класса:

### **-p rtpri**

Устанавливает для указанных процессов значение приоритета реального времени **rtpri**.

### **-t tqntm [-r res]**

Устанавливает квант времени для указанных процессов равным **tqntm**. Можно также дополнительно указать разрешение как описано далее.

При установке параметров для процессов с разделением времени допустимы следующие специфические опции класса:

### **-m tsuprilim**

Устанавливает верхний предел **tsuprilim** для значения пользовательского приоритета указанных процессов.

### **-p tsupri**

Устанавливает пользовательскому приоритету указанных процессов значение **tsupri**.

При установке параметров для интерактивных процессов допустимы следующие специфические опции класса:

### **-m iauprilim**

Устанавливает верхний предел **iauprilim** для значения пользовательского приоритета указанных процессов.

### **-p iaupri**

Устанавливает пользовательскому приоритету указанных процессов значение **iaupri**.

## **ИСПОЛЬЗОВАНИЕ**

### **Процессы реального времени**

Класс реального времени обеспечивает вытесняющее планирование с фиксированным приоритетом для процессов, которые должны срабатывать быстро и предсказуемо, позволяя приложению/пользователю полностью контролировать приоритеты планировщика. Если класс реального времени сконфигурирован в системе, он должен получить абсолютный контроль над верхним диапазоном приоритетов планировщика в системе. Это гарантирует, что готовый к выполнению процесс реального времени получит доступ к процессору раньше любого процесса, принадлежащего к любому другому классу.

Класс реального времени имеет диапазон значений приоритетов реального времени (**rtpri**), которые можно назначать процессам этого класса. Приоритеты реального времени задаются в диапазоне от 0 до **x**, причем, значение **x** конфигурируется и может быть выдано для системы, в которой уже сконфигурирован планировщик реального времени, с помощью команды:

```
priocntl -l
```

Правила планирования реального времени предполагают использование фиксированного приоритета. Приоритет планировщика для процесса реального времени никогда не меняется, - разве что, в результате явного запроса пользователя/приложения установить процессу приоритет **rtpri**.

Для процессов класса реального времени значение **rtpri**, во всех отношениях, эквивалентно приоритету планировщика для процесса. Значение **rtpri** полностью определяет приоритет процесса реального времени по отношению к другим процессам того же класса. Чем больше значение **rtpri**, тем более высокий приоритет получит процесс. Поскольку класс реального времени контролирует верхний диапазон приоритетов планировщика в системе, гарантируется, что готовый к выполнению процесс реального времени с максимальным значением **rtpri** всегда выбирается для запуска перед всеми остальными процессами в системе.

Помимо управления приоритетом, команда **priocntl** позволяет управлять продолжительностью кванта времени, выделяемого процессам реального времени. Значение кванта времени задает максимальный период времени, в течение которого процесс может работать, если только не закончится или не войдет в состояние ожидания ресурса или события (не "заснет"). Учтите, что если другой процесс с более высоким приоритетом станет пригодным к запуску, текущий работающий процесс может быть вытеснен прежде, чем отработает полный квант времени.

Команда:

```
priocntl -d [-i тип_идентификатора] [список_идентификаторов]
```

выдает приоритет реального времени и квант времени (в миллисекундах) для каждого процесса в наборе, задаваемом параметрами **тип\_идентификатора** и **список\_идентификаторов**.

Любое сочетание опций **-p** и **-t** можно задавать в командах **priocntl -s** или **priocntl -e** для процессов реального времени. Если опция не указана и процесс уже относится к классу реального времени, соответствующий параметр не меняется. Если опция не указана при переводе процесса в класс реального времени, соответствующий параметр получает стандартное значение. Стандартное значение **rtpri** - 0, а стандартное значение кванта времени зависит от значения **rtpri** и от конфигурации системы; см. **rt\_dptbl(4)**.

При использовании опции **-t tqntm**, можно дополнительно указать разрешение с помощью опции **-r res**. (Если разрешение не указано, предполагаются миллисекунды.) Если разрешение задается, оно должно быть положительным целым числом от 1 до 1000000000 и определяется как соответствующая доля секунды. Например, при указании **-t 10 -r 100** устанавливается разрешение в одну сотую секунды, а продолжительность получаемого кванта времени будет 10/100 секунды (одна десятая секунды). Хотя можно задавать очень детальные разрешения (до наносекунды), продолжительность кванта времени округляется до следующего целочисленного кратного разрешения системных часов. Запросы квантов времени 0 или

больших, чем (обычно очень большой) определяемый реализацией максимум, приводят к выдаче сообщения об ошибке.

Для перевода процесса в класс реального времени (из любого другого класса), пользователь, вызывающий команду **priocntl**, должен иметь привилегию суперпользователя. Для изменения значения приоритета (**rtpri**) или кванта времени для процесса реального времени, пользователь, вызывающий команду **priocntl**, должен либо быть суперпользователем, либо уже принадлежать к классу реального времени (командный интерпретатор пользователя должен быть процессом реального времени) и иметь реальный или эффективный идентификатор пользователя, совпадающий с реальным или эффективным идентификатором целевого процесса.

Приоритет реального времени и квант времени наследуются при выполнении системных вызовов **fork(2)** и **exec(2)**.

## Процессы с разделением времени

Правила планирования в классе процессов с разделением времени обеспечивают справедливое и эффективное распределение ресурсов процессора между процессами с различными потребностями в процессорном времени. Целью планирования процессов с разделением времени является обеспечение хорошего времени отклика для интерактивных процессов и хорошей производительности для заданий, требующих интенсивной работы процессора, обеспечивая при этом определенный контроль пользователя/приложения над планированием.

Класс процессов с разделением времени поддерживает диапазон значений пользовательского приоритета (**tsupri**), которые можно задавать для процессов класса. Пользовательские приоритеты могут задаваться в диапазоне от **-x** до **+x**, где значение **x** может конфигурироваться. Диапазон для конкретной системы можно получить с помощью команды:

```
priocntl -l
```

Цель установки пользовательского приоритета состоит в том, чтобы обеспечить определенную степень управления особенностями планирования процессов с разделением времени со стороны пользователя/приложения. Увеличение или уменьшение значения **tsupri** для процесса в классе разделения времени увеличивает или уменьшает приоритет процесса для планировщика. Не гарантируется, однако, что процесс с разделением времени и большим значением **tsupri** выполнится раньше, чем процесс с меньшим значением **tsupri**. Причина в том, что значение **tsupri** - только один фактор, используемый планировщиком при определении приоритета процесса с разделением времени. Система может динамически подстраивать внутренний приоритет планировщика процессов с разделением времени на основе других факторов, например, текущего использования процессора.

Кроме общесистемных ограничений пользовательских приоритетов (которые выдаются командой **priocntl -l**), есть и ограничение максимального пользовательского приоритета для каждого процесса (**tsuprilm**), которое задает максимальное значение **tsupri** для данного процесса.

Команда:

```
priocntl -d [-i тип_идентификатора] [список_идентификаторов]
```

выдает текущий пользовательский приоритет и максимальный пользовательский приоритет для каждого процесса с разделением времени в наборе, заданном параметрами **тип\_идентификатора** и **список\_идентификаторов**.

Любой процесс с разделением времени может понизить собственное значение **tsuprilim** (или значение для другого процесса с тем же идентификатором пользователя). Только процесс с разделением времени с привилегией суперпользователя может повысить значение **tsuprilim**. При переводе процесса в класс процессов с разделением времени, необходима привилегия суперпользователя для установки начального значения **tsuprilim** больше нуля.

Любой процесс с разделением времени может установить своему параметру **tsupri** (или этому параметру другого процесса с тем же идентификатором пользователя) любое значение, меньшее или равное значению **tsuprilim** для процесса. Попытки установить значение **tsupri** больше **tsuprilim** (и/или установить значение **tsuprilim** меньше **tsupri**) приводят к тому, что значение **tsupri** устанавливается равным **tsuprilim**.

Любая комбинация опций **-m** и **-p** может использоваться в командах **priocntl -s** или **priocntl -e** для процессов с разделением времени. Если опция не указана и процесс уже относится к классу процессов с разделением времени, соответствующий параметр обычно не меняется. Единственное исключение - когда опция **-p** не указана, а опция **-m** используется для установки значения **tsuprilim** меньше значения **tsupri**. В этом случае, значение **tsupri** устанавливается равным заданному значению **tsuprilim**. Если опция не указана при переводе процесса в класс процессов с разделением времени, соответствующий параметр получает стандартное значение. Стандартное значение **tsuprilim** - 0, а стандартное значение **tsupri** равно задаваемому значению **tsuprilim**.

Пользовательский приоритет и максимальное значение пользовательского приоритета для процессов с разделением времени наследуется при выполнении системных вызовов **fork(2)** и **exec(2)**.

## Интерактивные процессы

Правила планирования в классе интерактивных процессов обеспечивают справедливое и эффективное распределение ресурсов процессора между процессами с различными потребностями в процессорном времени при одновременной хорошей реакции на действия пользователя. Целью планирования процессов с разделением времени является обеспечение хорошего времени отклика для интерактивных процессов и хорошей производительности для заданий, требующих интенсивной работы процессора. Приоритеты интерактивных процессов можно менять аналогично приоритетам процессов с разделением времени, хотя измененные приоритеты будут настраиваться далее для обеспечения хорошей реакции на действия пользователя.

## ПРИМЕРЫ

Примеры для класса процессов реального времени представлены ниже:

### Пример 1: Перевод процессов в класс реального времени

В этом примере все процессы, выбираемые по заданному типу\_идентификатора и списку\_идентификаторов и не относящиеся к классу реального времени, переводятся в класс процессов реального времени с установкой приоритета реального времени 0. Приоритеты процессов, уже относящихся к классу реального времени, не меняются. Значение кванта времени для всех заданных процессов устанавливается равным 1/10 секунды.

```
example% priocntl -s -c RT -t 1 -r 10 -i тип_идентификатора список_идентификаторов
```

### Пример 2: Выполнение команды в реальном времени



В этом примере **команда** выполняется в классе реального времени с приоритетом реального времени 15 и квантом 20 миллисекунд:

```
example% priocntl -e -c RT -p 15 -t 20 команда
```

Примеры для класса процессов с разделением времени представлены ниже:

**Пример 3: Перевод процессов в класс с разделением времени**

В этом примере все процессы, выбираемые по заданному **типу\_идентификатора** и **списку\_идентификаторов** и не относящиеся к классу процессов с разделением времени, переводятся в класс процессов с разделением времени с установкой максимального и текущего пользовательского приоритета 0. Параметры процессов, уже работающих с разделением времени, не изменяются.

```
example% priocntl -s -c TS -i тип_идентификатора список_идентификаторов
```

**Пример 4: Выполнение команды в классе с разделением времени**

В этом примере **команда с аргументами** выполняется в классе процессов с разделением времени, с максимальным пользовательским приоритетом 0 и текущим пользовательским приоритетом -15:

```
example% priocntl -e -c TS -m 0 -p -15 команда [аргументы]
```

**СТАТУС ВЫХОДА**

Команда при завершении возвращает следующие значения:

Для опций **-d**, **-l** и **-s**:

- 0** Успешное выполнение.
- 1** Произошла ошибка.

Для опции **-e**:

Возврат статуса выхода выполненной команды свидетельствует об успешном срабатывании. Иначе:

- 1** Команду нельзя выполнить с указанным приоритетом.

**АТРИБУТЫ**

Описание следующих атрибутов см. на странице справочного руководства **attributes(5)**:

ТИП АТРИБУТА	ЗНАЧЕНИЕ АТРИБУТА
Доступен в пакете	SUNWcsu
CSI	Включен

**ССЫЛКИ**



## ДИАГНОСТИКА

Команда **priocntl** выдает следующие сообщения об ошибках:

### **Process(es) not found**

Ни один из указанных процессов не существует.

### **Specified processes from different classes**

Для установки параметров используется опция **-s**, опция **-с класс** не указана, и заданы процессы, принадлежащие к разным классам.

### **Invalid option or argument**

Передана недопустимая или неизвестная опция или аргумент.

Последнее изменение: 10 января 2000 года

Copyright (no c) - Fuck copyright!, 2001-2003 В. Кравчук, [OpenXS Initiative](#), перевод на русский язык