


ХОЧУ ПОМОЧЬ ПРОЕКТУ

Контроль и ограничение потребления системных ресурсов ОС



besteq.ru

РЕКЛАМА

Этикетировочная машина BESTEQ-HALS. Аналог ОБ-КЭТ-С2

Собственный сервис • Обучение персонала • Европейское качество

Узнать цену

Статья / Контроль и ограничение потребления системных ресурсов ОС

Модуль resource предоставляет базовые механизмы для измерения и контроля системных ресурсов, используемых процессом. Он доступен на платформах WebAssembly wasm32-emscripten и wasm32-wasi. Дополнительную информацию о функциях модуля resource в форме WebAssembly.

Для указания определенных системных ресурсов и для запроса информации об использовании либо текущего процесса, либо его дочерних элементов используются [СИМВОЛЬНЫЕ КОНСТАНТЫ](#).

Использование ресурсов может быть ограничено с помощью функции [resource.setrlimit\(\)](#), описанной ниже. Каждый ресурс контролируется парой ограничений: мягким ограничением и жестким ограничением. Мягкий предел - это текущий предел, который может быть снижен или повышен процессом с течением времени. Мягкое ограничение никогда не может превышать жесткое ограничение. Жесткий предел может быть снижен до любого значения, превышающего мягкое ограничение, но не повышен.

Внимание! Только процессы с эффективным UID суперпользователя могут повысить жесткое ограничение.

Конкретные ресурсы, которые могут быть ограничены, зависят от системы. Они описаны на справочной странице `$ man getrlimit` (команда Linux). Перечисленные ниже ресурсы поддерживаются, если их поддерживает базовая операционная система. Ресурсы, которые не могут быть проверены или проконтролированы операционной системой, не определены в этом модуле для этих платформ.

Содержание:

- `resource.RLIM_INFINITY` [предел неограниченного ресурса](#),
- `resource.getrlimit()` [получает текущие мягкое и жесткое ограничение ресурса](#),
- `resource.setrlimit()` [устанавливает новые лимиты потребления ресурса](#),
- `resource.prlimit()` [объединяет setrlimit\(\) и getrlimit\(\) в одну функцию](#),
- [Пример ограничения потребления ресурсов процессора и памяти](#).

`resource.RLIM_INFINITY`:

Константа `resource.RLIM_INFINITY`, которая используется для представления предела неограниченного ресурса.

`resource.getrlimit(resource)`:

Функция `resource.getrlimit()` возвращает кортеж `(soft, hard)` с текущими мягкими и жесткими ограничениями ресурса.

Вызывает [исключение ValueError](#), если указан недопустимый ресурс, или вызывает исключение [исключение OSError](#), если базовый системный вызов неожиданно завершается с ошибкой.

`resource.setrlimit(resource, limits)`:

Функция `resource.setrlimit()` устанавливает новые лимиты потребления ресурса, который передается аргументу `resource`.

Аргумент `limits` должен быть кортежем `(soft, hard)` из двух целых чисел, описывающих новые лимиты. Значение [resource.RLIM_INFINITY](#) может использоваться для запроса на потребление неограниченного ресурса.

Метод вызывает [исключение ValueError](#), если указан недопустимый ресурс, если новое мягкое ограничение превышает жесткое ограничение или если процесс пытается повысить свое жесткое ограничение.

Указание ограничения [resource.RLIM_INFINITY](#), когда жесткое или системное ограничение для этого ресурса является ограниченным, приведет к ошибке ValueError. Процесс с действующим UID суперпользователя может запросить любое допустимое значение ограничения, включая неограниченное, но ValueError все равно будет поднят, если запрошенный предел превышает установленный системой.

РЕКЛАМА


Яндекс Маркет

0+

Выгодный комплект электроинструментов

С кешбэком баллами Плюса

6%



Makita

Функция `resource.setrlimit(resource, limit)` также может вызвать ошибку, если базовый системный вызов не работает.

Вызывает `resource.setrlimit` с аргументами `resource`, `limit`.

Рекомендуется использовать функцию `resource.setrlimit(resource[, limits])`:

Функция `resource.setrlimit(resource[, limits])` объединяет [resource.setrlimit\(\)](#) и [resource.getrlimit\(\)](#) в одну функцию и поддерживает ресурсы произвольного процесса PID.

Если `pid` не задан, то применяется к текущему процессу. Аргументы `resource` и `limits` имеют то же значение, что и в `resource.getrlimit()`, за исключением того, что значение аргумента `limits` не является обязательным.

Когда `pid` задан, то функция возвращает лимит ресурсов процесса `pid`. Когда задан, то функция устанавливает лимит ресурсов процесса, и возвращается прежний лимит ресурсов.

Вызывает `resource.setrlimit` с аргументами `pid`, `resource`, `limits`.

Случай, когда невозможно найти `pid`, и [PermissionError](#), когда у пользователя нет доступа к процессу.

Вызывает [событие аудита](#) `resource.prlimit` с аргументами `pid`, `resource`, `limits`.

Доступность: Linux >= 2.6.36 with glibc >= 2.13.

Пример ограничения потребления ресурсов процессора и памяти.

Если лень оптимизировать память программы или корректировать работу процессора, то можно просто установить лимиты:

```
import signal
import resource

def time_exceeded(signo, frame):
    print("CPU exceeded...")
    raise SystemExit(1)

def set_max_runtime(seconds):
    soft, hard = resource.getrlimit(resource.RLIMIT_CPU)
    resource.setrlimit(resource.RLIMIT_CPU, (seconds, hard) )
    signal.signal(signal.SIGXCPU, time_exceeded)

def set_max_memory(size):
    soft, hard = resource.getrlimit(resource.RLIMIT_AS)
    resource.setrlimit(resource.RLIMIT_AS, (size, hard))
```

Здесь можно увидеть две опции: установку на максимальное процессорное время и максимальный предел используемой памяти.

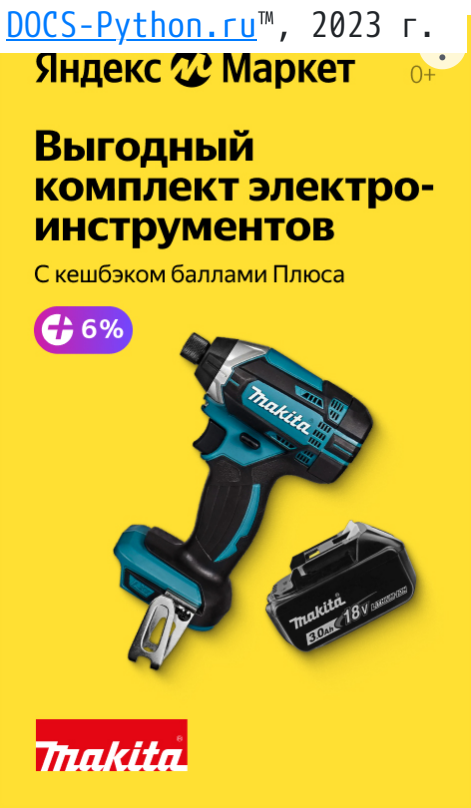
При ограничении работы процессора необходимо извлечь мягкий и жёсткий лимиты для конкретного ресурса (`resource.RLIMIT_CPU`), а затем установить его значение. В качестве мягкого ограничения используется количество секунд `seconds` указанное в аргументе функции, и ранее полученное жёсткое ограничение. В конце функции необходимо зарегистрировать сигнал, который будет отвечать за выход, если мягкое ограничение процессорного времени превышено.

Что касается памяти, то, как и в случае с процессором, устанавливаем мягкий и жёсткий лимиты. Для установки ограничений используется функция [resource.setrlimit\(\)](#) со значением `size` и жёсткое ограничение, которое было получено.

Содержание раздела:

- [КРАТКИЙ ОБЗОР МАТЕРИАЛА.](#)
- [Символьные константы модуля resource](#)

Вверх



(Внимание! При копировании материала ссылка на источник обязательна)

[@docs_python_ru](#)