



ХОЧУ ПОМОЧЬ ПРОЕКТУ

Модуль faulthandler в Python, дампинг трассировок

 mango-office.ru

РЕКЛАМА



Виртуальная АТС Расширенная

До 120 функций. Когда нужно распределять звонки по группам сотрудников и по типам...

Узнать больше

Статьи

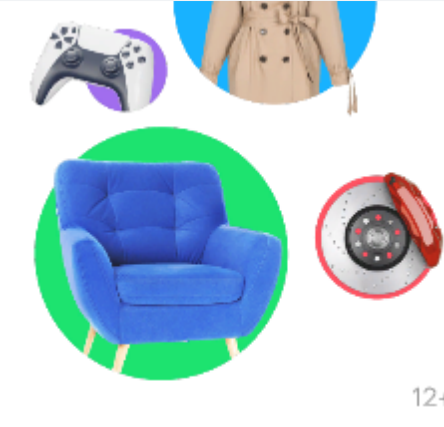
Вывод

Модуль

Сигналы

Выход

СИ



12+

/ Модуль faulthandler в Python, дампинг трассировок

ировки программы Python

функции для явного дампинга трассировок Python, в случае сбоя, после тайм-аута или по [.\),](#) чтобы установить обработчики ошибок для сигналов SIGSEGV, SIGFPE, SIGABRT, SIGBUS и их при запуске, установив переменную среды [PYTHONFAULTHANDLER](#) или используя [параметр командной строки -X faulthandler](#).

Обработчик сбоев совместим с обработчиками системных сбоев, такими как Apport в Unix системах или обработчик сбоев Windows. Если доступна функция sigaltstack(), то модуль использует альтернативный стек для обработчиков сигналов. Это позволяет выгружать трассировку даже при переполнении стека.

Обработчик сбоев вызывается в катастрофических случаях и поэтому может использовать только сигнально-безопасные функции (например, он не может выделять память в куче). Из-за этого ограничения дамп трассировки минимален по сравнению с обычными трассировками Python:

- Поддерживается только ASCII. При кодировании используется обработчик ошибок обратной косой черты.
- Каждая строка ограничена 500 символами.
- Отображаются только имя файла, имя функции и номер строки. (без исходного кода)
- Он ограничен 100 кадрами и 100 потоками.
- Порядок обратный: самый последний вызов отображается первым.

По умолчанию трассировка Python записывается в [sys.stderr](#). Чтобы увидеть трассировку, приложения должны быть запущены в терминале. Файл журнала можно также передать в функцию [faulthandler.enable\(\)](#).

Модуль реализован на языке C, поэтому трассировки могут быть сброшены при сбое или когда Python находится в зависшем состоянии.

Режим разработки Python вызывает функцию faulthandler.enable() при запуске Python.

Пример использования модуля faulthandler:

Пример ошибки сегментации в Linux с включением обработчика ошибок и без него:

```
# без обработчика ошибок
$ python3 -c "import ctypes; ctypes.string_at(0)"
Segmentation fault

# включаем обработчик ошибок
$ python3 -q -X faulthandler
>>> import ctypes
>>> ctypes.string_at(0)
Fatal Python error: Segmentation fault

Current thread 0x00007fb899f39700 (most recent call first):
  File "/home/python/cpython/Lib/ctypes/__init__.py", line 486 in string_at
  File "<stdin>", line 1 in <module>
Segmentation fault
```

Вверх

Содержание раздела:

- [КРАТКИЙ ОБЗОР МАТЕРИАЛА.](#)
- РЕКЛАМА [включение/выключение обработчика ошибок модуля faulthandler](#)
- [Выгрузка трассировки стека при обработке ошибок faulthandler](#)