

proc(1)

НАЗВАНИЕ

proc, pflags, pcred, pmap, pldd, psig, pstack, pfiles, pwdx, pstop, prun, pwait, ptree, ptime - средства работы с файловой системой **proc**

СИНТАКСИС

```
/usr/bin/pflags [ -r ] [ идентификатор_процесса | core ] ...
/usr/bin/pcred [ идентификатор_процесса | core ] ...
/usr/bin/pmap [ -rxLF ] [ идентификатор_процесса | core ] ...
/usr/bin/pldd [ -F ] [ идентификатор_процесса | core ] ...
/usr/bin/psig идентификатор_процесса ...
/usr/bin/pstack [ -F ] [ идентификатор_процесса | core ] ...
/usr/bin/pfiles [ -F ] идентификатор_процесса ...
/usr/bin/pwdx [ -F ] идентификатор_процесса ...
/usr/bin/pstop идентификатор_процесса ...
/usr/bin/prun идентификатор_процесса ...
/usr/bin/pwait [ -v ] идентификатор_процесса ...
/usr/bin/ptree [ -a ] [ [ идентификатор_процесса | пользователь ]... ]
/usr/bin/ptime команда [ аргумент ... ]
```

ОПИСАНИЕ

Средства работы с файловой системой **proc** - это утилиты, использующие возможности, предоставляемые файловой системой **/proc** (см. **proc(4)**). Большинство из них воспринимает список идентификаторов процессов; они же воспринимают в качестве идентификатора процесса также значение **/proc/nnn**, так что шаблон командного интерпретатора **/proc/*** можно использовать для указания всех процессов в системе. Некоторые из средств работы с **proc** можно также применять к файлам **core** (образы памяти процессов, см. **core(4)**); те, что позволяют обрабатывать эти файлы, могут обрабатывать список либо идентификаторов процессов, либо имен файлов, либо и тех, и других.

pflags

Выдает флаги трассировки **/proc**, ожидающие и удерживаемые сигналы, и другую информацию о состоянии **/proc** для каждого легковесного процесса (потока, **Iwp**) в каждом процессе.

pcred

Выдает привилегии (эффективные, реальные и сохраненные идентификаторы пользователя и группы) для каждого процесса.

pmap

Выдает карту адресного пространства для каждого процесса.

pldd

Выдает список динамически подключаемых библиотек для каждого процесса, включая разделяемые объекты, явно подключенные с помощью вызова **dlopen(3DL)**. См. **ldd(1)**.

psig

В _дает обработчики сигналов для каждого процесса. См. **signal(3HEAD)**.

pstack

Выдает содержимое стека в шестнадцатеричном и символьном виде для каждого легковесного процесса (**Iwp**) в каждом процессе.

pfiles

Выдает информацию **fstat(2)** и **fcntl(2)** для всех открытых файлов в каждом процессе.

pwdx

Выдает текущий рабочий каталог каждого процесса.

pstop

Останавливает каждый процесс (**PR_REQUESTED stop**).

prun

Снова запускает каждый (остановленный) процесс (обратное действие по отношению к **pstop**).

pwait

Ждет завершения всех указанных процессов.

ptree

Выдает деревья процессов, содержащих процессы с указанными идентификаторами процессов или идентификаторами пользователей так, что порожденные процессы сдвигаются вправо по отношению к соответствующим родительским процессам. Аргумент, состоящий исключительно из цифр, рассматривается как идентификатор процесса, - в противном случае аргумент считается регистрационным именем пользователя. По умолчанию выдаются все процессы.

ptime

Выдает время выполнения команды, как утилита **time(1)**, но на базе учета микросостояния для получения предсказуемой точности. В отличие от **time(1)**, время работы процессов-потомков команды не учитывается.

ОПЦИИ

Поддерживаются следующие опции:

-r

(Только для **pflags**) Если процесс остановлен, выдавать значения регистров процессора для него.

-r

(Только для **pmap**) Выдавать зарезервированные адреса процесса.

-x

(Только для **pmap**) Выдавать особенности сопоставления адресов в резидентных/разделяемых/приватных областях.

-l

(Только для **pmap**) Выдавать не разрешенные динамическим компоновщиком имена.

-a

(Только для **ptree**) Выдавать все процессы, включая потомков процесса 0.

-v

(Только для **pwait**) Выдавать детальную информацию; выдавать в стандартный выходной поток коды завершения процессов.

-F

Выполнять безусловно; перехватывать целевой процесс даже если им управляет другой процесс.

ИСПОЛЬЗОВАНИЕ

Следующие средства работы с файловой системой **proc** останавливают целевые процессы по ходу их изучения и выдачи результатов: **pfiles**, **pldd**, **pmap** и **pstack**. Пока процесс остановлен, он ничего делать не может. Так, например, если X-сервер проверяется одной из этих утилит, работающих в окне под

управлением этого X-сервера, вся оконная система может оказаться заблокированной, поскольку утилита будет пытаться выдавать результаты в окно, которое не может быть обновлено. Взаимную блокировку в таком случае можно снять из другой системы с помощью утилиты [rlogin\(1\)](#), прекратив работу утилиты.

Надо быть осторожным при использовании флага [-F](#). Применение двух управляющих процессов к одной "жертве" может вызвать хаос. Безопасность гарантирована только если главный управляющий процесс, обычно, - отладчик, остановил процесс-жертву и ничего не делает в тот момент, когда к [Л](#)тому процессу применяется соответствующая утилита для работы с файловой системой [proc](#).

Некоторые средства работы с файловой системой [proc](#) можно также применять к файлам [core](#), как показано в описании синтаксиса в начале страницы. Файл [core](#) - моментальный снимок состояния процесса, который создается ядром перед прекращением работы процесса при обработке сигнала или утилитой [gcore\(1\)](#). Некоторым средствам работы с файловой системой [proc](#) может потребоваться определить имя выполняемого файла, соответствующего процессу, образ которого сброшен, или имена разделяемых библиотек, связанных с этим процессом. Эти файлы необходимы, например, чтобы получить информацию таблицы символов для [pstack\(1\)](#). Если соответствующее средство работы с файловой системой [proc](#) не может найти необходимый выполняемый файл или разделяемую библиотеку, часть символьной информации выдать оно не сможет. Аналогично, если файл [core](#) из одной версии операционной системы проверяется на другой версии, интерфейс редактирования связей времени выполнения (run-time link-editor debugging interface - [librtld_db](#)) может не удастся проинициализировать. В этом случае символьная информация для разделяемых библиотек будет недоступна.

СТАТУС ВЫХОДА

Утилиты возвращают следующие значения выхода:

0 Успешное выполнение.
не ноль Произошла ошибка.

ФАЙЛЫ

/proc/*

файлы процесса

/usr/proc/lib/*

Вспомогательные файлы для средств работы с файловой системой [proc](#)

АТРИБУТЫ

Описание следующих атрибутов см. на странице справочного руководства [attributes\(5\)](#):

ТИП АТРИБУТА	ЗНАЧЕНИЕ АТРИБУТА
Доступен в пакете	SUNWcsu (32-bit)
	SUNWesxu (64-bit)

ССЫЛКИ

[gcore\(1\)](#), [ldd\(1\)](#), [ps\(1\)](#), [pwd\(1\)](#), [rlogin\(1\)](#), [time\(1\)](#), [truss\(1\)](#), [wait\(1\)](#), [fcntl\(2\)](#), [fstat\(2\)](#), [dlopen\(3DL\)](#), [core\(4\)](#), [proc\(4\)](#), [attributes\(5\)](#), [signal\(3HEAD\)](#)

Последнее изменение: 17 ноября 1999 года

