NAME

core - memory image file format

SYNOPSIS

#include <sys/param.h>

DESCRIPTION

A small number of signals which cause abnormal termination of a process also cause a record of the process's in-core state to be written to disk for later examination by one of the available debuggers. (See sigaction(2).) This memory image is written to a file named by default **programname.core** in the working directory; provided the terminated process had write permission in the directory, and provided the abnormality did not cause a system crash. (In this event, the decision to save the core file is arbitrary, see savecore(8).)

The maximum size of a core file is limited by the RLIMIT_CORE setrlimit(2) limit. Files which would be larger than the limit are not created.

With a large limit, a process that had mapped a very large, and perhaps sparsely populated, virtual memory region, could take a very long time to create core dumps. The system ignores all signals sent to a process writing a core file, except SIGKILL which terminates the writing and causes immediate exit of the process. The behavior of SIGKILL can be disabled by setting tunable sysctl(8) variable *kern.core_dump_can_intr* to zero.

The name of the file is controlled via the sysctl(8) variable *kern.corefile*. The contents of this variable describes a filename to store the core image to. This filename can be absolute, or relative (which will resolve to the current working directory of the program generating it).

The following format specifiers may be used in the *kern.corefile* sysctl to insert additional information into the resulting core filename:

%H	Machine hostname.
%I	An index starting at zero until the sysctl *debug.ncores* is reached. This can be useful for limiting the number of corefiles generated by a particular process.
%N	process name.
%P	processes PID.
%S	signal during core.
%U	process UID.

The name defaults to *%N.core*, yielding the traditional FreeBSD behaviour.

By default, a process that changes user or group credentials whether real or effective will not create a

corefile. This behaviour can be changed to generate a core dump by setting the sysctl(8) variable *kern.sugid_coredump* to 1.

Corefiles can be compressed by the kernel if the following item is included in the kernel configuration file:

```
options GZIO
```

The following sysctl control core file compression:

kern.compress_user_cores	Enable compression of user cores. A value of 1 configures gzip(1) compression, and a value of 2 configures zstd(1) compression. Compressed core files will have a suffix of '.gz' or '.zst' appended to their filenames depending on the selected format.
kern.compress_user_cores_level	Compression level. Defaults to 6.

NOTES

Corefiles are written with open file descriptor information as an ELF note. By default, file paths are packed to only use as much space as needed. However, file paths can change at any time, including during core dump, and this can result in truncated file descriptor data.

All file descriptor information can be preserved by disabling packing. This potentially wastes up to PATH_MAX bytes per open fd. Packing is disabled with

```
sysctl kern.coredump_pack_fileinfo=0.
```

Similarly, corefiles are written with vmmap information as an ELF note, which contains file paths. By default, they are packed to only use as much space as needed. By the same mechanism as for the open files note, these paths can also change at any time and result in a truncated note.

All vmmap information can be preserved by disabling packing. Like the file information, this potentially wastes up to PATH_MAX bytes per mapped object. Packing is disabled with

```
sysctl kern.coredump_pack_vmmapinfo=0.
```

EXAMPLES

In order to store all core images in per-user private areas under */var/coredumps*, the following sysctl(8) command can be used:

```
sysctl kern.corefile=/var/coredumps/%U/%N.core
```

SEE ALSO

gdb(1) (*ports/devel/gdb*), gzip(1), kgdb(1) (*ports/devel/gdb*), setrlimit(2), sigaction(2), sysctl(8)

HISTORY

A **core** file format appeared in Version 1 AT&T UNIX.