NAME

crash – examine system images

SYNOPSIS

/usr/sbin/crash [**–d** *dumpfile*] [**–n** *namelist*] [**–w** *output-file*]

AVAILABILITY

SUNWcsu

DESCRIPTION

The **crash** command is used to examine the system memory image of a running or a crashed system by formatting and printing control structures, tables, and other information. Command line arguments to **crash** are *dumpfile*, *namelist*, and *output-file*.

OPTIONS

–d *dumpfile* Specify *dumpfile* as the file containing the system memory image. The default *dumpfile* is **/dev/mem**. The system image can also be the pathname of a dump file generated by the **savecore**(1M) utility.

–n *namelist* Specify the text file *namelist* which contains the symbol table information needed for symbolic access to the system memory image to be examined. The default *namelist* is **/dev/ksyms**. Note: It is recommended that **crash** dumps be analyzed on a machine having the same kernel architecture as the machine that the dump was taken from.

–w *output-file* When the **crash** command is invoked, a session is initiated. The output from a **crash** session is directed to *output-file*. The default *output-file* is the standard output.

USAGE

Input during a **crash** session is of the form:

function [*argument*...]

where *function* is one of the **crash** functions described in the **Functions** subsection of this manual page, and *arguments* are qualifying data that indicate which items of the system image are to be printed.

The default for process-related items is the current process for a running system or the process that was running at the time of the crash for a crashed system. Similarly, the default for thread-related items is the current thread for a running system or the thread that was running at the time of the crash for a crash system. If the contents of a table are being dumped, the default is all active table entries.

Function Options

The following function options are available to **crash** functions wherever they are semantically valid. Valid function options are shown in **Functions**.

–e Display every entry in a table.

–f Display the full structure.

–p Interpret all address arguments in the command line as physical addresses. If the addresses specified are not physical addresses, results are inconsistent.

–s *process* Specify a process slot other than the default.

–w *filename* Redirect the output of a function to *filename*.

The functions **mode**, **defproc**, and **redirect** correspond to the function options −**p**, −**s**, and −**w**. The **mode** function may be used to set the address translation mode to physical or virtual for all subsequently entered functions; **defproc** sets the value of the process slot argument for subsequent functions; and **redirect** redirects all subsequent output.

Output from **crash** functions may be piped to another program in the following way:

function [*argument*. . .] **!** *shell_command*

For example,

mount ! grep rw

writes all mount table entries with an **rw** flag to the standard output. The redirection option (−**w**) cannot be used with this feature.

Depending on the context of the function, numeric arguments are assumed to be in a specific radix. Counts are assumed to be decimal. Addresses are always hexadecimal. Table address arguments larger than the size of the function table are interpreted as hexadecimal addresses; those smaller are assumed to be decimal slots in the table. Default bases on all arguments may be overridden. The C conventions for designating the bases of numbers are recognized. A number that is usually interpreted as decimal is interpreted as hexadecimal if it is preceded by **0x** and as octal if it is preceded by **0**. Decimal override is designated by **0d**, and binary by **0b**.

Aliases for functions may be any uniquely identifiable initial substring of the function name. Traditional aliases of one letter, such as **p** for **proc**, remain valid.

Many functions accept different forms of entry for the same argument. Requests for table information accept a table entry number, a physical address, a virtual address, a symbol, a range, or an expression. A range of slot numbers may be specified in the form *a*−*b* where *a* and *b* are decimal numbers. An expression consists of two operands and an operator. An operand may be an address, a symbol, or a number; the operator may be **+**, −, ∗, **/**, **&**,or **|**. An operand that is a number should be preceded by a radix prefix if it is not a decimal number (**0** for octal, **0x** for hexadecimal, **0b** for binary). The expression must be enclosed in parentheses. Other functions accept any of these argument forms that are meaningful.

Two abbreviated arguments to **crash** functions are used throughout. Both accept data entered in several forms. They may be expanded into the following:

table_entry = *slot number* | *address* | *symbol* | *range* | *expression*

start_addr = *address* | *symbol* | *expression*

Functions

? [−**w** *filename*]
List available functions.

!*command* Escape to the shell and execute *command*.

as [–**e**] [–**f**] [–**l**] [–**p**] [–**w** *filename*] [*proc*...]
Print information on process address space. If the –**l** is specified, all locks relevant to the address space structure are also displayed.

base [–**w** *filename*] *number*...
Print *number* in binary, octal, decimal, and hexadecimal. A number in a radix other than decimal should be preceded by a prefix that indicates its radix as follows: **0x**, hexadecimal; **0**, octal; and **0b**, binary.

buffer [–**w** *filename*] [*–format*] *bufferslot*

buffer [–**w** *filename*] [*–format*] [–**p**] *start_addr*
Alias: **b**
Print the contents of a buffer in the designated format. The following *format* designations are recognized: –**b**, byte; –**c**, character; –**d**, decimal; –**x**, hexadecimal; –**o**, octal; and, –**i**, inode. If no format is given, the previous format is used. The default format at the beginning of a **crash** session is hexadecimal.

bufhdr [–**f**] [–**w** *filename*] [[–**p**] *table_entry*...]
Alias: **buf**
Print system buffer headers.

callout [–**l**] [–**w** *filename*]
Alias: **c**
Print the **callout** table. If the –**l** option is specified, the contents of the locks pertaining to the **callout** structure are also displayed.

class [–**w** *filename*] [*table_entry*...]
Print information about process scheduler classes.

cpu [–**w** *filename*] *start_addr*...
Display the **cpu** structure pointed to by *start_addr*.

ctx [–**w** *filename*] *start_addr*
SPARC: Display the context structure given its starting address.

defproc [–**w** *filename*] [–**c**] [–**r**]

defproc [–**w** *filename*] [*slot*] [–**r**]
Set the value of the process slot argument. The process slot argument may be set to the current process' slot number (–**c**) or the slot number may be specified. If the –**r** option is specified, the process slot number and the current thread pointer are reset (the process slot number is set to the current process' slot number and the current thread pointer is set to the currently running thread). If no argument is entered, the value of the previously set slot number is printed. At the start of a **crash** session, the process slot is set to the current process.

defthread [−**p**] [−**w** *filename*] [−**r**] [−**c** *thread_addr*]
With no argument **defthread** displays the address of the **current thread**. If −**c** is specified, the **current thread** is set to point to the currently running thread. If an address is specified the **current thread** is specified to be that thread. If −**r** is specified, the **default process** is also set to be the process pertaining to **current thread**.

dispq [−**w** *filename*] [*table_entry*. . .]
Print the dispatcher (scheduler) queues.

ds [−**w** *filename*] *virtual_address*. . .
Alias: **ts**
Print the data symbol whose address is closest to, but not greater than, the address entered.

file [−**e**] [−**f**] [−**w** *filename*] [[−**p**] *table_entry*. . .]
Alias: **f**
Print the **file** table.

findaddr [−**w** *filename*] *table slot*
Print the address of *slot* in *table*. Only tables available to the **size** function are available to **findaddr**.

findslot [−**w** *filename*] *virtual_address*. . .
Print the table, entry slot number, and offset for the address entered. Only tables available to the **size** function are available to **findslot**.

fs [−**f**] [−**w** *filename*] [[−**p**] *table_entry*. . .]
Alias: **vfssw**
Print the filename system information table.

gdp [−**e**] [−**f**] [−**l**] [−**w** *filename*] [[−**p**] *table_entry* ...]
x86: Print the gift descriptor protocol table. If the −**l** option is specified, display all relevant lock information.

hat [−**a**] [−**w** *filename*] *start_addr*. . .
Display the **hat** structure pointed to by *start_addr*. If the −**a** option is specified, the chain of **hat** structures is displayed.

help [−**w** *filename*] *function*. . .
Print a description of the named function, including syntax and aliases.

hment [−**a**] [−**w** *filename*] *start_addr*. . .
Display the **hment** structure pointed to by *start_addr*. If the −**a** option is specified, the chain of **hment** structures is displayed.

kfp [−**w** *filename*] [*thread_addr*]
Print the kernel frame pointer (**kfp**) for the start of a kernel stack trace. If *thread_addr* is specified the **kfp** for that thread is printed.

kmastat [−**w** *filename*]
Print kernel memory allocator statistics.

kmausers [−**e**] [−**f**] [−**w** *filename*] [*cache*name. . .]

Print the information about the medium and large users of the kernel memory allocator that have current memory allocations. The output consists of one entry for each unique stack trace specifying the total amount of memory and number of allocations that was made with that stack trace.

This function is only available if the kernel has the **KMF_AUDIT** flag set in **kmem_flags**. (See the **NOTES** below.)

If one or more cache names (e.g. kmem_alloc_256) are specified the scan of memory usage is restricted to those caches. By default all caches are included.

If the (−**e**) option is used, the small users of the allocator are included. The small users are allocations that total less than 1024 bytes of memory or for which there are less than 10 allocations with the same stack trace.

If the (−**f**) option is used, the stack traces are printed for each individual allocation.

lck [−**e**] [−**w** *filename*] [[−**p**] *lock_addr*. . .]

Alias: **l**

Print record locking information. If the −**e** option is used or lock address arguments are given, the record lock list is printed. If no argument is entered, information on locks relative to UFS inodes is printed.

linkblk [−**e**] [−**w** *filename*] [[−**p**] *table_entry*. . .]

Print the **linkblk** table.

lwp [−**w** *filename*] [*lwp_addr*. . .]

Display the **lwp** structure pointed to by *lwp_addr*. If no address is specified as an argument, the **lwp** structure associated with the **current thread** is displayed.

major [−**w** *filename*] [*entry*. . .]

Print the **MAJOR** table.

map [−**w** *filename*] *mapname*. . .

Print the **map** structure of the given *mapname*.

mblk [−**e**] [−**f**] [−**w** *filename*] [[−**p**] *table_entry*. . .]

Print allocated streams message block and data block headers.

mblkusers [−**e**] [[−**f**] [−**w** *filename*]

Print the information about the medium and large users of the stream message memory allocator that have current mblk allocations. The output consists of one entry for each unique stack trace specifying the total amount of memory and number of allocations that was made with that stack trace.

This function is only available if the kernel has the **KMF_AUDIT** flag set in **kmem_flags**. (See the **NOTES** below.)

If the (−**e**) option is used, the small users of the allocator are included. The small users are allocations that total less than 1024 bytes of memory or for which there are less than 10 allocations with the same stack trace.

If the (−**f**) option is used, the stack traces are printed for each individual allocation.

mode [−**w** *filename*] [*mode*]
Set address translation of arguments to virtual (**v**) or physical (**p**) mode. If no mode argument is given, the current mode is printed. At the start of a **crash** session, the mode is virtual.

mount [−**f**] [−**w** *filename*] [[−**p**] *table_entry*. . .]
Alias: **m**, **vfs**
Print information about mounted filename systems.

mutex [−**w** *filename*] *mutex_addr*. . .
Dump the contents of the **mutex** pointed to by *mutex_addr*.

mutextable [−**w** *filename*]
Display every entry in the **mutex_init_table**.

nfsnode [−**w** *filename*] [*node_addr*]
Print the NFS remote nodes.

nm [−**w** *filename*] *symbol*. . .
Print value and type for the given symbol.

od [−**p**] [−**w** *filename*] [*−format*] [*−mode*] [−**s** *process*] *start_addr* [*count*]
Alias: **rd**
Print *count* values starting at *start_addr* in one of the following formats: character (−**c**), decimal (−**d**), hexadecimal (−**x**), octal (−**o**), ASCII (−**a**), or hexadecimal/character (−**h**), and one of the following modes: long (−**l**), short (−**t**), or byte (−**b**). The default mode for character and ASCII formats is byte; the default mode for decimal, hexadecimal, and octal formats is long. The format −**h** prints both hexadecimal and character representations of the addresses dumped; no mode needs to be specified. When format or mode is omitted, the previous value is used. At the start of a **crash** session, the format is hexadecimal and the mode is long. If no count is entered, 1 is assumed.

page [−**e**] [−**l**] [−**w** *filename*] [[−**p**] *table_entry*. . .]
Print information about pages. If −**l** is specified, display all relevant locking information.

pcb [−**w** *filename*] [*thread_addr*]
Print the process control block. If no arguments are given, the active **pcb** for the current thread is printed. If *thread_addr* is specified, the **pcb** associated with that thread is printed.

pcfsnode [−**w** *filename*] [*node_address*]
Print the PCFS filename and directory node tables.

pmgrp [−**l**] [−**w** *filename*] *start_addr*. . .
Display the **pmgrp** structure pointed to by *start_addr*. If the −**l** option is specified, the chain of **pmgrp** structures is displayed.

prnode [−**e**] [−**f**] [−**l**] [−**w** *filename*] [[−**p**] *table_entry*. . .]
Print information about the private data of processes being traced. If −**l** is specified, all relevant locking information is also displayed.

proc [−**e**] [−**f**] [−**l**] [−**w** *filename*] [[−**p**] [−**a**] *table_entry*. . . | *#procid*. . .]
proc [−**e**] [−**f**] [−**l**] [−**w** *filename*] [−**r**]
Alias: **p**
Print the process table. Process table information may be specified in two ways. First, any mixture of table entries and process IDs may be entered. Each process ID must be preceded by a **#**. Alternatively, process table information for runnable processes may be specified with the runnable option (−**r**). If the −**l** option is specified, all relevant locking information is displayed.

pte [−**w** *filename*] *start_addr*. . .
Display the **pte** structure pointed to by *start_addr*.

pty [−**e**] [−**f**] [−**w** *filename*] [−**s**] [−**h**] [−**l**] [−**t** *type*] [[−**p**] *table_entry*]
Print the pseudo ttys presently configured. The −**l**, −**h** and −**s** options give information about the STREAMS modules **ldterm**, **ptem** and **pckt**, respectively. If the −**t** option is used, the table for the single **pty** type specified is printed.

qrun [−**w** *filename*]
Print the list of scheduled STREAMS queues.

queue [−**e**] [−**f**] [−**w** *filename*] [[−**p**] *table_entry*. . .]
Print the allocated STREAMS queues.

quit Alias: **q**
Terminate the **crash** session. Exit.

redirect [−**w** *filename*] [−**c**]
redirect [−**w** *filename*] [*newfile*]
Used with a *filename*, redirects output of a **crash** session to *newfile*. If no argument is given, the filename name to which output is being redirected is printed. Alternatively, the close option (−**c**) closes the previously set filename and redirects output to the standard output.

rtdptbl [−**w** *filename*] [*table_entry*. . .]
Print the real-time dispatcher (scheduler) parameter table. See **rt_dptbl**(4).

rtproc [−**w** *filename*]
Print information about processes in the real-time scheduler class.

rwlock [−**w** *filename*] *rw_addr*. . .
Dump the contents of the **rwlock** (read/write lock) structure pointed to by *rw_addr*.

search [−**w** *filename*] [−**m** *mask*] [−**s** *process*] *pattern* [−**p**] *start_addr length*
Print the words in memory that match *pattern*, beginning at the *start_addr* for *length* words. The mask is AND'ed (&) with each memory word and the result compared against the pattern. The mask defaults to **0xffffffff**.

sema [–**w** *filename*] *sema_addr*...
Dump the contents of the semaphore structure pointed to by *sema_addr*.

size [–**x**] [–**w** *filename*] [*structure_name*...]
Print the size of the designated structure. The –**x** option prints the size in hexadecimal. If no argument is given, a list of the structure names for which sizes are available is printed. The following example lists those structures:

> size

buf	**callout**	**flckinfo**	**fifonode**
filock	**flox**	**lwp**	**pp**
prnode	**proc**	**snode**	**thread**
tty	**ufs_inode**	**user**	**vfs**
vfssw	**vnode**		
datab	**dblk**	**linkblk**	
mblk	**msgb**	**queue**	
stdata	**streams**		

>

sment [–**w** *filename*] *start_addr*...
Display the **sment** structure pointed to by *start_addr*.

smgrp [–**l**] [–**w** *filename*] *start_addr*...
Display the **smgrp** structure pointed to by *start_addr*. If the –**l** option is specified, the chain of **smgrp** structures is displayed.

snode [–**e**] [–**f**] [–**l**] [–**w** *filename*] [[–**p**] *table_entry*...]
Print information about open special filenames. If the –**l** option is specified, all relevant locking information is also displayed.

stack [–**w** *filename*] [–**u**] [–**p**] [*thread*]

stack [–**w** *filename*] [–**k**] [–**p**] [*thread*]
Alias: **s**
Dump the stack. The –**u** option prints the user stack. The –**k** option prints the kernel stack. If no arguments are entered, the kernel stack for the current thread is printed. Otherwise, the kernel stack for the currently running thread is printed.

status [–**w** *filename*]
Print system statistics.

stream [–**e**] [–**f**] [–**w** *filename*] [[–**p**] *start_addr*...]
Print the STREAMS table.

strstat [–**w** *filename*]
Print STREAMS statistics.

trace [–**w** *filename*] [[–**p**] [*thread_addr*]]
Alias: **t**
Print stack trace. Displays the stack trace of the currently running or the specified thread.

thread [−**e**] [−**f**] [−**l**] [−**w** *filename*] *slot number*
With no option, displays the **current thread**. If −**e** is specified, all threads in the system are displayed. Otherwise all threads pointed to by the addresses specified as argument are displayed.

ts [−**w** *filename*] *virtual_address*. . .
Alias: **ds**
Print text symbol closest to the designated address.

tsdptbl [−**w** *filename*] [*table_entry*. . .]
Print the time-sharing dispatcher parameter table. See **ts_dptbl**(4).

tsproc [−**w** *filename*]
Print information about processes in the time-sharing dispatcher parameter table (scheduler class).

tty [−**e**] [−**f**] [−**l**] [−**w** *filename*] [−**t** *type* [[−**p**] *table_entry*. . .]]

tty [−**e**] [−**f**] [−**l**] [−**w** *filename*] [[−**p**] *start_addr*]
Valid types: **pp**, **iu**.
Print the tty table. If no arguments are given, the tty table for both tty types is printed. If the −**t** option is used, the table for the single tty type specified is printed. If no argument follows the type option, all entries in the table are printed. A single tty entry may be specified using *start_addr*. The −**l** option prints the line discipline information.

uinode [−**d**] [−**e**] [−**f**] [−**l**] [−**r**] [−**w** *filename*] [[−**p**] *table_entry*. . .]
Alias: **ui**
Print the UFS inode table. The −**d** option will list the address and i-number of all UFS inodes in use and on the free list. If the −**l** option is specified, all relevant locking information is also displayed. The −**r** option will display all free UFS inodes.

user [−**e**] [−**f**] [−**l**] [−**w** *filename*] [*process*]
Alias: **u**
Print the user structure for the designated process. If the −**l** option is specified, display all relevant locking information.

var [−**w** *filename*]
Alias: **v**
Print the tunable system parameters.

vfs [−**e**] [−**w** *filename*] [[−**p**] *address*. . .]
Alias: **m**, **mount**
Print information about mounted filename systems.

vfssw [−**f**] [−**w** *filename*] [[−**p**] *table_entry*. . .]
Alias: **fs**
Print information about configured filename system types.

vnode [−**w** *filename*] [−**l**] [−**p**] *vnode_addr*. . .]
Print information about vnodes.

vtop [−**w** *filename*] [−**s** *process*] *start_addr*. . .
Print the physical address translation of the virtual address *start_addr*.

EXIT STATUS

The following exit values are returned:

0 Success.

1 An error has occurred.

FILES

/dev/mem system image of currently running system

/dev/ksyms system namelist

SEE ALSO

savecore(1M), **ts_dptbl**(4)

NOTES

Kernel core dumps should be examined on the same platform they were created on.

The following steps are necessary in order to set **KMF_AUDIT** in **kmem_flags** (which is needed for the **kmausers** and **mblkusers** commands):

```
ok boot kadb -d
kadb: kernel/unix
kadb[0]: startup:b
kadb[0]: :c
kadb[0]: kmem_flags/W1
kadb[0]: :c
```