NAME

kdb − kernel debugger

SYNOPSIS

kdb [-] [namelist]

or

kdb [core [namelist]]

DESCRIPTION

Kdb is a debugging package for the kernel layer of the **MERT** operating system. Like the **UNIX** debugger, *kdb* is used to examine core image files. Typically, the file will be either a core image of a system resulting from a crash or /dev/mem when examining the current operating system.

The command line has two forms:

> If a minus (-) is specified, *kdb* will use /dev/mem as the core file. References to segments which are not currently in memory will be satisfied by reading the segment from /dev/swap.
>
> If a core file is specified, *kdb* will treat all references to the segments which are not in the core image as errors.

If no arguments are given, the default core file is *kore* in the current directory and the default namelist is /mrt/krn.sym.

The format of *kdb* requests is a one or two character mneumonic followed by a list of parameters. Numeric parameters are assumed to be octal unless terminated by a decimal point. In the following description only the first two characters of mneumonics are required:

$

> The user, supervisor, and kernel stack pointers, followed by the general registers are typed. This command causes *kdb* to set its internal tables for virtual address mapping of the kernel address space to that at the time the kore image was produced. This should not be used when debugging /dev/mem.

! command line

> The exclamation point (!) is stripped and the rest of the line is sent to the shell for execution.

*

> The values of the kernel segmentation registers (sdr and sar) are displayed.

= symbol

> The value of the symbol **symbol** is displayed.

dct pn1 pn2

> The Dispatcher Control Tables starting at process **pn1** through **pn2** are displayed. If **pn1** is not entered, all the DCTs are typed.

dl pri1 pri2 0 < pri1 <=7 pri1 <=pri2 <=7

> The Dispatcher Control Tables (DCT) entries which are on the processor priority chain *pri1* are listed in the order inwhich they appear on the linked list. After all DCT entries are listed, *pri1* is incremented. If *pri1* is less than *pri2,* the next chain is listed.

ml

> The Segment Discriptor Entries (SDE) are listed in order of increasing memory.

msg

> The contents of all the message buffers are displayed.

pmsg pn1 pn2

The contents of all messages on the message queues of processes **pn1** through **pn2** are displayed.

rsde segid

The RSDE (Resident Segment Descriptor Entry) for the segment *segid* is displayed. If *segid* is not specified, all RSDEs are displayed.

sde address

The SDE (Segment Descriptor Entry) pointed to by *address* is displayed. If no address is specified all SDEs are displayed.

seg segid offset n

The contents of the segment *segid* starting at byte offset *offset* into the segment and continuing for *n* bytes is displayed. If *n* is not specified, the end of the segment is assumed. If *offset* is not specified, zero is assumed.

scan start end pattern mask

Kernel virtual address space is searched starting at *start* up to and including *end* for a match on the pattern *pattern.* Each word of kernel memory is masked with *mask* before the comparisons are made.

snap addr[d] n

The kernel virtual address *addr* and *n* consecutive bytes are typed. If the address is followed by a "d" (no blanks or tabs) the address is interpreted as a D-space address, otherwise I-space is assumed. Virtual address to file offset is done via the image of the segmentation registers in *kdb's* internal tables. If /dev/mem is being examined, there is no way to get the current setting of the segmentation registers. If a core file is being examined, the '$' command will set *kdb's* mapping tables to those of the system at the time the core image was produced.

xt pn

Extracts an image of the process *pn* into the file korexxx, where xxx is the octal index into the DCT tables for the process. The program *sdb* can be used to examine these core images.

ALSO SEE

sdb(e),kdmp(e),tdmp(e)

DIAGNOSTICS

"Open err:" if a file cannot be opened, otherwise "?".

FILES

kore core image file
/dev/mem
/dev/swap
/mrt/krn.sym namelist