NAME

limit, ulimit, unlimit – set or get limitations on the system resources available to the current shell and its descendents

SYNOPSIS

/usr/bin/ulimit [–**f**] [*blocks*]

sh

ulimit [–[**HS**] [**a**| **cdfnstv**]]
ulimit [–[**HS**]] **c**| **d**| **f**| **n**| **s**| **t**| **v**]] *limit*

csh

limit [–**h**] [*resource* [*limit*]]
unlimit [–**h**] [*resource*]

ksh

ulimit [–**HSacdfnstv**] [*limit*]

DESCRIPTION

/usr/bin/ulimit

The **ulimit** utility sets or reports the file-size writing limit imposed on files written by the shell and its child processes (files of any size may be read). Only a process with appropriate privileges can increase the limit.

sh

The Bourne shell built-in function, **ulimit**, prints or sets hard or soft resource limits. These limits are described in **getrlimit**(2).

If *limit* is not present, **ulimit** prints the specified limits. Any number of limits may be printed at one time. The –**a** option prints all limits.

If *limit* is present, **ulimit** sets the specified limit to *limit*. The string **unlimited** requests the largest valid limit. Limits may be set for only one resource at a time. Any user may set a soft limit to any value below the hard limit. Any user may lower a hard limit. Only a super-user may raise a hard limit; see **su**(1M).

The –**H** option specifies a hard limit. The –**S** option specifies a soft limit. If neither option is specified, **ulimit** will set both limits and print the soft limit.

The following options specify the resource whose limits are to be printed or set. If no option is specified, the file size limit is printed or set.

–**c**	maximum core file size (in 512-byte blocks)
–**d**	maximum size of data segment or heap (in kbytes)
–**f**	maximum file size (in 512-byte blocks)
–**n**	maximum file descriptor plus 1
–**s**	maximum size of stack segment (in kbytes)
–**t**	maximum CPU time (in seconds)
–**v**	maximum size of virtual memory (in kbytes)

csh

The C-shell built-in function, **limit**, limits the consumption by the current process or any process it spawns, each not to exceed *limit* on the specified *resource*. If *limit* is omitted, print the current limit; if *resource* is omitted, display all limits. (Run the **sysdef**(1M) command to obtain the maximum possible limits for your system. The values reported are in hexidecimal, but can be translated into decimal numbers using the **bc**(1) command).

–h Use hard limits instead of the current limits. Hard limits impose a ceiling on the values of the current limits. Only the privileged user may raise the hard limits.

resource is one of:

cputime Maximum CPU seconds per process.
filesize Largest single file allowed; limited to the size of the filesystem. (see **df**(1M)).
datasize (heapsize) Maximum data size (including stack) for the process. This is the size of your virtual memory (see **swap**(1M)).
stacksize Maximum stack size for the process. (see **swap**(1M)).
coredumpsize Maximum size of a core dump (file). This limited to the size of the filesystem.
descriptors Maximum number of file descriptors. (run **sysdef()**).
memorysize Maximum size of virtual memory.

limit is a number, with an optional scaling factor, as follows:

*n***h** Hours (for **cputime**).
*n***k** *n* kilobytes. This is the default for all but **cputime**.
*n***m** *n* megabytes or minutes (for **cputime**).
*mm***:***ss* Minutes and seconds (for **cputime**).

unlimit removes a limitation on *resource*. If no *resource* is specified, then all resource limitations are removed. See the description of the **limit** command for the list of resource names.

–h Remove corresponding hard limits. Only the privileged user may do this.

ksh The Korn shell built-in function, **ulimit**, sets or displays a resource limit. The available resources limits are listed below. Many systems do not contain one or more of these limits. The limit for a specified resource is set when *limit* is specified. The value of *limit* can be a number in the unit specified below with each resource, or the value **unlimited**. The **H** and **S** flags specify whether the hard limit or the soft limit for the given resource is set. A hard limit cannot be increased once it is set. A soft limit can be increased up to the value of the hard limit. If neither the **H** or **S** options is specified, the limit applies to both. The current resource limit is printed when *limit* is omitted. In this case the soft limit is printed unless **H** is specified. When more that one resource is specified, then the limit name and unit is printed before the value.

–a Lists all of the current resource limits.
–c The number of 512-byte blocks on the size of core dumps.
–d The number of K-bytes on the size of the data area.
–f The number of 512-byte blocks on files written by child processes (files of any size may be read).
–n The number of file descriptors plus 1.
–s The number of K-bytes on the size of the stack area.
–t The number of seconds (CPU time) to be used by each process.

–v The number of K-bytes for virtual memory.

If no option is given, **–f** is assumed.

OPTIONS

The following option is supported by **ulimit**:

–f Set (or report, if no *blocks* operand is present), the file size limit in blocks. The **–f** option is also the default case.

OPERANDS

The following operand is supported by **ulimit**:

blocks The number of 512-byte blocks to use as the new file size limit.

EXAMPLES

/usr/bin/ulimit

To limit the stack size to 512 kilobytes:

```
% ulimit -s 512
% ulimit -a
% time(seconds)         unlimited
file(blocks)            100
data(kbytes)            523256
stack(kbytes)           512
coredump(blocks)        200
nofiles(descriptors)    64
memory(kbytes)          unlimited
```

sh/ksh

To limit the number of file descriptors to 12:

```
$ ulimit -n 12

$ ulimit -a
time(seconds)           unlimited
file(blocks)            41943
data(kbytes)            523256
stack(kbytes)           8192
coredump(blocks)        200
nofiles(descriptors)    12
vmemory(kbytes)         unlimited
```

csh

To limit the size of a core dump file size to 0 kilobytes:

```
% limit coredumpsize 0
% limit
cputime        unlimited
filesize       unlimited
datasize       523256 kbytes
stacksize      8192 kbytes
coredumpsize   0 kbytes
descriptors    64
memorysize     unlimited
```

To remove the above limitation for the core file size:

```
% unlimit coredumpsize
% limit
cputime          unlimited
filesize         unlimited
datasize         523256 kbytes
stacksize        8192 kbytes
coredumpsize     unlimited
descriptors      64
memorysize       unlimited
```

ENVIRONMENT

See **environ**(5) for descriptions of the following environment variables that affect the execution of **ulimit**: **LC_CTYPE**, **LC_MESSAGES**, and **NLSPATH**.

EXIT STATUS

The following exit values are returned by **ulimit**:

0 Successful completion.

>0 A request for a higher limit was rejected or an error occurred.

SEE ALSO

bc(1), **csh**(1), **ksh**(1), **sh**(1), **su**(1M), **df**(1M), **swap**(1M), **sysdef**(1M), **getrlimit**(2)