`config.h` File [3]

Note: It is strongly recommended that you use the UNICOS installation and configuration menu system (ICMS) to maintain your system configuration, rather than manually editing this file. (For more information on the ICMS, see the online help files associated with the tool and *UNICOS System Configuration Using ICMS*, Cray Research publication SG–2412.) Typically, this file is stored in `/usr/src/uts/cf.`*xxxx*`/config.h`.

This chapter summarizes parameters found in the `/usr/src/uts/c1/cf/config.h` file.

Note: The default values shown reflect the settings used for initial installations; appropriate values for upgrades will be changed automatically as part of the upgrade conversion process. Sites performing an upgrade should only need to change values manually if they are enabling features or changing hardware.

These parameters pertain to general configuration, the UNICOS multilevel security (MLS) feature, and TCP/IP. To set the general configuration and TCP/IP parameters, use the following menu:

```
Configure System
    ->Kernel Configuration
```

To set the MLS parameters, use the following menu:

```
Configure System
    ->Multilevel Security (MLS) Configuration
```

In general, a value of `0` disables a parameter and a value of `1` enables it.

Table 1 through Table 3 summarize resource and configuration parameters for the UNICOS kernel. Some parameters are not available in the ICMS and may therefore be edited only by those sites with a source license.

Table 1. Kernel parameters in config.h (common)

Parameter	Default value	Description
CDLIMIT	010000000000	Maximum file size in blocks (see ulimit(2)).
DMODE	1	Default offline file retrieval mode for sites running data migration. 1 indicates automatic retrieval; 0 indicates manual-only retrieval.
EXTDCORE	0	Extended core file naming.
FLUSHONPANIC	1	The capability to flush buffered data to target I/O devices before the system is halted.
LDCHCORE	0	Memory clicks reserved for ldcache blocks. These are used for ldcache blocks specified as type MEM. This value can be changed at boot time in the CSL parameter file.
LINK_MAX	1000	Maximum number of directory entries and maximum number of links to one file.
MAXASYN	192	Maximum number of asynchronous I/O structures allowed per process. MAXASYN has considerable performance impact on applications utilizing asynchronous I/O (for example, listio and aqio system calls).
MAXPIPE	20	Maximum number of blocks allocated per pipe on the pipe device.
MAXRAH	8	Maximum number of read-ahead blocks per read operation.
NASYN	800	Number of asynchronous I/O headers.
NBLK_FCTR	20	Factor for maximum request size to/from system buffer cache. The default of 20 means that 1/20th (5%) of the system buffer cache can be allocated to a single request.
NBUF_FCTR	20	Number of cache blocks based on memory size. The default of 20 means that 1/20th (5%) of the memory is assigned to buffer cache. This can be changed at boot time to an absolute number of cache blocks with the NBUF parameter in the CSL parameter file.

Parameter	Default value	Description
NCALL	NPROC	Number of callout table entries.
NCLIST	1000	Maximum number of clists. These are terminal I/O buffers.
NCRBUF	4	Maximum number of checkpoint and restart buffers.
NC1INODE	NINODE	Number of in-core inodes for NC1FS (file system migration parameter). Must be equal to NINODE. This parameter is not in the ICMS.
NC_NAMLEN	15	Maximum size of path-name component that will be cached in the directory name look-up cache (DNLC).
NC_SIZE	1024	Number of entries in directory name look-up cache. This should be less than NINODE.
NEXECS	4	Maximum number of parallel exec operations with maximum arguments. This parameter is not in ICMS.
NFILE	2100	Maximum number of files that can be open at one time system-wide.
NFLOCKS	100	Number of file-lock regions.
NHBUF_FCTR	4	Number of hash entries in the hash table for system cache, based on number of blocks assigned to system cache. The default of 4 specifies 1 hash list for every 4 system buffers.
NINODE	1500	Number of unique files that can be open at one time. Also the size of the in-core inode table.
NLDCH	0 or 2000	Number of SSD logical device cache headers, 0 for CRAY J90 systems and 2000 for all other supported mainframes. This can be changed at boot time in the CSL parameter file.
NLDMAP	250	Number of file systems open concurrently.
NMNT	250	Number of slice structures.
NMOUNT	75	Number of file systems (local, NFS, DFS, and so on) that can be mounted.

Parameter	Default value	Description
NPBUF	500	Number of physical I/O headers. This can be changed at boot time in the CSL parameter file.
NPLCHCTL	0	Number of partition cache headers. This can be changed at boot time in the CSL parameter file. This parameter is necessary only if you specify the -M CMEM option of the pcache(8) command.
NPROC	650	Maximum number of processes that may be active simultaneously.
NPTY	128	Number of pseudo terminals (tty/pty pairs). This is also the maximum number of active terminal sessions.
NQUOTA	1400 or 0	If your site is running file system quotas, 1400 is the default number of in-core quota entries; otherwise, it is set to 0.
NSESS	300	Maximum number of sessions open simultaneously.
NSIDEBUF	48	Number of blocks for the SSD side door buffer. All sites with an SSD should have a side door buffer assigned.
NSU_ASYN	4	Number of asynchronous I/O headers reserved for the system.
NTEXT	100	Maximum number of shared-text programs that can be running simultaneously.
NUSERS	200	Maximum number of users defined for the fair share scheduler.
PLCHCORE	0	Memory clicks reserved for partition cache blocks. These are used for partition cache blocks specified as type MEM. This can be changed at boot time in the CSL parameter file.
TAPE_MAX_PER_DEV	65536	Maximum number of bytes per device for buffered I/O.

Parameter	Default value	Description
TAPE_MAX_CONF_UP	4	Maximum number of tapes that may be configured up.
XTRASEC	3	Number of CPU seconds that a process or session may use following receipt of a SIGCPULIM signal before the SIGKILL signal is sent to terminate the process or session.

Table 2. Kernel parameters in config.h (common) not in ICMS

Parameter	Default value	Description
ACNICE	HZ	The lowest process nice value is saved as pc_acnice after the specified number of seconds of CPU time.
FSLG_BUFSIZE	400	Number of file system log records.
K_OPEN_MAX	16384	Maximum number of open files per process.
KSTACKSIZE	1500 or 2000	Kernel stack size, 2000 if you have the DCE Distributed File Service (DFS) installed, 1500 if you do not.
	⚠	**Caution:** Do not change this value. This parameter does not appear in ICMS because it should not be changed.
MAX_UNLINKED_BYTES	26214400	Maximum bytes allowed in unlinked files for checkpoint and restart.
NC1MINRAW	20	Multiplier for automatic mixed buffer.
NC1INODE	NINODE	Number of in-core inodes for NC1FS (file system migration parameter). Must be equal to NINODE.
NEXECS	4	Maximum number of parallel exec operations with maximum arguments.
U_MAXPACK	16	Maximum number of outstanding user packets allowed. For CRAY J90 and CRAY J90se systems only.

Table 3. Kernel parameters in config.h (CRAY J90 systems only)

Parameter	Default value	Description
NMT	16	Number of non-tape daemon tapes.
U_MAXDEV	16	Number of user devices.
U_MAXPACK	16	Number of outstanding user packets allowed.

Table 4 lists the dynamic kernel memory parameters.

Table 4. Dynamic kernel memory parameters

Parameter	Default value	Description
KM_CHM_PADSZ	(((KSTACKSIZE + KM_WPU-1) >> KM_WPU_SHIFTC) * (NPROC/2))	Size of the memory padding. The total size (in KM_UNITS) must be large enough for NPROC/2 kernel stack entries. Compilation of lowmem.c will fail if this constraint is not met.
KM_EXPAND_UNITS	128	The number of units each expansion will add. Because expansion space is acquired from coremap, it must be a multiple of coremap units. That is: (KM_EXPAND_UNITS * KM_WPU) == multiple of (MEMKLIK*NWPC)
KM_NO_THRASH	(4*KM_EXPAND_UNITS)	The number of units that must be reached before a contraction will take place. This prevents contractions from reclaiming expansion space that may be requested within a short time span. Expansions can be expensive if they require shuffling memory.

Parameter	Default value	Description
		An expansion may require moving the bitmap, which is stored in dynamic kernel memory, in order to increase its capacity. The frequency of such moves is dependent upon the values of KM_WPU and KM_EXPAND_UNITS. For example, if KM_EXPAND_UNITS is 512, each expansion requires 8 additional words of bitmap memory to manage the space. If KM_WPU is 16, this will cause the bitmap to be reallocated and moved every two expansions. If KM_WPU is 128, the bitmap will need to be reallocated every 8 expansions. In a given configuration of 640 initial units, 128 words per unit, and 128 units acquired with each expansion, the bitmap will not move until the 120th expansion; that is, 640 units divided by 64 bits per word results in 10 words required being initially required for the bitmap. However, since the bitmap is also allocated in an area of memory equal to the value of KM_WPU, 128 words are reserved for the bitmap. This leaves 118 extra words in the bitmap unit. Since each expand of 128 units requires an additional 2 words, 59 expands can be done before the bitmap capacity is exceeded and the bitmap will be grown (and moved if necessary).
KM_UNITS	1408	Initial dynamic kernel memory units from coremap.
	⚠	**Caution:** The value of (KM_UNITS * KM_WPU) should be a multiple of (MEMKLIK*NWPC) to avoid wasting space acquired from coremap. That is, at boot time, a chunk of memory is acquired from coremap to be managed at a finer granularity.
KM_WPU	128	Words per unit. This value must be a power of 2.
KM_WPU_SHIFTC	7	Number of shifts to multiply or divide by the KM_WPU value.

Table 5 lists the parameters in config.h for the CRAY T90 series shared memory feature configuration. These parameters are significant only for the CRAY T90 series.

Table 5. Shared memory kernel parameters in config.h

Parameter	Default value	Description
SHMMAX	128	Maximum size in clicks of a shared segment. Because memory is allocated in MEMKLIK units, this should be a multiple of MEMKLIK. This value must be greater than or equal to 0.
SHMMIN	1	Minimum size in clicks of a shared segment. This value must be greater than or equal to and less than the SHMMAX value.
SHMMNI	20	Maximum number of shared memory identifiers available concurrently in the system (that is, the size of the shmem table). This value must be greater than 0.
SHMSEG	2	Maximum number of shared segments that a process can have attached simultaneously. This value must be 0, 1, or 2. (The upper bound is a hardware restriction.)

Table 6 lists the parameters in config.h that are used for setting process error thresholds. These values represent the number of allowable errors of each type (program range, operand range, and error exit) per connect to a CPU. The counters are reset on each pass through resume(). The default value is 500 for all systems.

If the values are set too high, the counts may never be reached (especially on a busy system), and the failing process will not be aborted. The definition of unreachable will vary according to application and system load. Setting the values to 0 disables this feature.

Table 6. Kernel parameters in config.h for process error thresholds

Parameter	Default value	Description
MAXUSRERR	500	Maximum number of error exits that may be encountered before the process is aborted. Setting the value to disables this feature.
MAXUSRORE	500	Maximum number of operand range errors that may be encountered before the process is aborted. Setting the value to disables this feature.
MAXUSRPRE	500	Maximum number of program range errors that may be encountered before the process is aborted. Setting the value to 0 disables this feature.

Table 7 lists the parameters in config.h that are used for interprocess (IPC) semaphore defines and message defines.

Table 7. IPC kernel parameters in config.h

Parameter	Default value	Description
MSGMAX	2048	Maximum size of a message in bytes. This value must be greater than 0 and less than the MSGMNB value.
MSGMNB	4096	Maximum number of bytes that may be in a message queue. This value must be greater than and less than 999999.
MSGMNI	40	Maximum number of message queues that may be in the system at one time (that is, the size of the msgque table). This value must be greater than 0.
MSGSEG	1024	Number of message segments. The size of the message area is MSGSEG * MSGSSZ bytes. (This will be rounded up to MEMKLIK.) This value must be greater than or equal to the MSGTQL value. The upper limit is 9999.

Parameter	Default value	Description
MSGSSZ	32	Message segment size. The message area is split up into segments of size MSGSSZ bytes. Each message is allocated the number of segments required to hold the message. This value must be greater than and less than the MSGMAX value, and it must be a multiple of 8 bytes.
MSGTQL	100	Number of message headers in the system. There is one message header per active message. This value must be greater than or equal to the MSGMNI value. The upper limit is 9999.
SEMAEM	16384	Maximum adjust-on-exit value that can be set on a semop(2) system call by specifying the SEM_UNDO flag. This value must be greater than 0 and less than the SEMVMX value.
SEMMNI	40	Maximum number of semaphore sets that may be in the system at one time (that is, the size of the sema table). This value must be greater than and less than 1000.
SEMMNS	100	Number of semaphores in the system (that is, the size of the sem table). A semaphore set may consist of one or more semaphores. This value must be greater than or equal to the SEMMNI value.
SEMMNU	40	Number of undo structures in the system (that is, the size of the semu table). One entry is used by each process that registers semaphore operations to be undone at process termination. This value must be greater than and less than the NPROC value. A value close to the SEMMNI value will conserve table space. (The undo structure size in words is 3 * SEMUME + 2.)
SEMMSL	25	Maximum number of semaphores per set. This value must be greater than 0 and less than the SEMMNI value.

Parameter	Default value	Description
SEMOPM	25	Maximum number of operations per semop(2) system call. Operations can be performed on the semaphores within one set. This value must be greater than 0 and less than or equal to the SEMMSL value.
SEMUME	20	Maximum number of undo entries per process. Each undo entry in the undo structure allows for an undo operation on one semaphore in a semaphore set. This value must be greater than 0 and less than or equal to the SEMMNS value. To conserve table space, set SEMUME to a value less than the SEMMNI value. (The undo structure size in words is 3 * SEMUME + 2.)
SEMVMX	32767	Maximum value of an individual semaphore. This value must be greater than and less than 2**32.

Table 8 lists the parameters in config.h for the UNICOS multilevel security (MLS) feature. These parameters are significant only if your site enables the UNICOS MLS feature. See *General UNICOS System Administration*, Cray Research publication SG–2301, for more information on setting these parameters and on the UNICOS MLS feature.

Note: In general, a value of 0 disables a parameter and a value of 1 enables it.

Table 8. MLS kernel parameters in config.h

Parameter	Default value	Description
COMPART_ACTIVE_DEFAULT	0	Generic MLS defaults.
COMPART_VALID_DEFAULT	0	Generic MLS defaults.
CONSOLE_MSG	0	Capability to send a message to /dev/console when any user exceeds MAXLOGS login attempts.
DECLASSIFY_DISK	0	Deactivates declassification of the disk; activates when the value is not 0.
DECLASSIFY_PATTERN	0	Declassifies disk write pattern.

Parameter	Default value	Description
DELAY_MULT	0	Multiplier for failed logins. When set to 1 (enabled), the next login attempt is delayed by (# *of failed logins*) * (LOGDELAY) seconds. Otherwise, the delay period is not multiplied by any factor, and the next login is delayed LOGDELAY seconds.
DEV_ENFORCE_ON	0	Capability to enforce strict device labeling.
DISABLE_ACCT	0	Disables the login account when MAXLOGS is exceeded. Used with DISABLE_TIME.
DISABLE_TIME	-1	Duration in seconds for which the user is disabled when the MAXLOGS is exceeded. If –1 (or any negative value), the user is disabled indefinitely.
FORCED_SOCKET	NORMAL	When set to NORMAL, syslogd(8) can use sockets and pipes.
FSETID_RESTRICT	1	If nonzero, only a privileged process can create setuid / setgid files.
LOGDELAY	0	Maximum number of seconds to delay between login attempts.
MAXLOGS	0	Maximum number of failed login attempts allowed before a user is disabled if the DISABLE_ACCT is set to 1.
MAXSLEVEL	0	Maximum security level allowed on the system.
MINSLEVEL	0	Minimum security level allowed on the system.
MLS_INTEGRITY	NORMAL	Integrity code in secure.c. This setting is not in the ICMS.
MLS_OBJ_RANGES	NORMAL	Allow object ranges outside the system label range in mount and setdevs.
NFS_REMOTE_RW_OK	1	Enables (nonzero) or disables (zero) NFS-mounting of a remote file system in read-write mode.
NFS_SECURE_EXPORT_OK	1	Enables (nonzero) or disables (zero) exporting a secure file system with NFS.
OVERWRITE_COUNT	3	Declassifies disk overwrite count.
PASS_MAXSIZE	8	Maximum size for machine-generated passwords.

Parameter	Default value	Description
PASS_MINSIZE	8	Minimum size for machine-generated passwords.
PRIV_SU	1	When set to 1, root (UID 0) has privilege.
RANDOM_PASS_ON	0	When set to yes (1), machine-generated passwords are enabled for all users. Setting to no (0) disables this.
SANITIZE_PATTERN	0	The pattern used to scrub disks if SECURE_SCRUB is enabled. The default pattern is 0.
SECURE_NET_OPTIONS	(NETW_SOCK_COMPAT | NETW_RCMD_COMPAT)	Options for running TCP/IP on a secure system. The value may be any combination of the following: • NETW_STRICT_B1, in which strict B1 evaluation rules are applied by the system; therefore, TCP sessions are restricted to a single label, regardless of the type of remote host. • NETW_SOCK_COMPAT, in which sockets are automatically made multilevel if the creating process has PRIV_SOCKET. • NETW_RCMD_COMPAT, in which traditional hosts.equiv and .rhosts behavior is allowed. If NETW_RCMD_COMPAT is not used, remote logins with rlogin(1) to root are disallowed and all other logins must be to the same user name and require the same user ID on both systems. The default value of represents the combination of NETW_SOCK_COMPAT and NETW_RCMD_COMPAT. To add the strict B1 evaluation rules, you should add the NETW_STRICT_B1 value; that is, (NETW_SOCK_COMPAT | NETW_RCMD_COMPAT | NETW_STRICT_B1).
SECURE_MAC	0	When set to 1, enforces system high/system low MAC.
SECURE_OPERATOR_CONSOLE	"/dev/console"	Secure console for use by the MLS operator. This parameter is unused.

Parameter	Default value	Description
SECURE_PIPE	NORMAL	Enables/disables "read down" on pipes.
SECURE_SCRUB	NORMAL	Enables (SECURE) or disables (NORMAL) scrubbing of data blocks on file deletion.
SECURE_SYSTEM_CONSOLE	""	Secure console for MLS administration. This parameter is unused.
SLG_ACT_NFS	SLGOFF	Enables (SLGON) or disables (SLGOFF) logging of NFS activity.
SLG_ACT_NQS	SLGOFF	Enables (SLGON) or disables (SLGOFF) logging of NQS activity.
SLG_ALL_NAMI	SLGOFF	Enables (SLGON) or disables (SLGOFF) logging of all mkdir, rmdir, link, and rm calls.
SLG_ALL_RM	SLGOFF	Enables (SLGON) or disables (SLGOFF) logging of all remove requests.
SLG_ALL_VALID	SLGOFF	Enables (SLGON) or disables (SLGOFF) logging of all access requests.
SLG_BUFSIZE	163840	Size of the security log buffer. The default value holds approximately 1000 security log records. Must be a multiple of 4.
SLG_CF_NET	SLGOFF	Enables (SLGON) or disables (SLGOFF) logging of network configuration changes.
SLG_CF_NQS	SLGOFF	Enables (SLGON) or disables (SLGOFF) logging of NQS configuration changes.
SLG_CF_UNICOS	SLGON	Enables (SLGON) or disables (SLGOFF) logging of UNICOS configuration changes.
SLG_DIR	/usr/adm/sl	Full path name of directory where security logs are kept.
SLG_DISCV	SLGON	Enables (SLGON) or disables (SLGOFF) logging of discretionary access violations.
SLG_FILE	slogfile	File name of active security log.
SLG_FILEXFR	SLGON	Enables (SLGON) or disables (SLGOFF) logging of all file transfers.
SLG_FPREFIX	s.	Prefix of retired security logs.

Parameter	Default value	Description
SLG_JEND	SLGON	Enables (SLGON) or disables (SLGOFF) logging of job ends.
SLG_JSTART	SLGON	Enables (SLGON) or disables (SLGOFF) logging of job starts.
SLG_LINKV	SLGON	Enables (SLGON) or disables (SLGOFF) logging of all link violations.
SLG_LOG_AUDIT	SLGON	Enables (SLGON) or disables (SLGOFF) logging of audit criteria changes.
SLG_LOG_CHDIR	SLGON	Enables (SLGON) or disables (SLGOFF) logging of all chdir(1) requests.
SLG_LOG_CRL	SLGOFF	Enables (SLGON) or disables (SLGOFF) logging of Cray/REELlibrarian activity.
SLG_LOG_DAC	SLGON	Enables (SLGON) or disables (SLGOFF) logging of discretionary access control (DAC) activities.
SLG_LOG_IPNET	SLGON	Enables (SLGON) or disables (SLGOFF) logging of Internet Protocol (IP) layer activity.
SLG_LOG_OPER	SLGON	Enables (SLGON) or disables (SLGOFF) logging of operator actions.
SLG_LOG_PRIV	SLGOFF	Enables (SLGON) or disables (SLGOFF) logging of all privileges used in system calls.
SLG_LOG_SECSYS	SLGON	Enables (SLGON) or disables (SLGOFF) logging of non-inode security system calls.
SLG_LOG_SHUTDWN	SLGON	Enables (SLGON) or disables (SLGOFF) logging of system shutdown.
SLG_LOG_STARTUP	SLGON	Enables (SLGON) or disables (SLGOFF) logging of system startup.
SLG_LOG_TAPE	SLGOFF	Enables (SLGON) or disables (SLGOFF) logging of tape activity.
SLG_LOG_TCHG	SLGON	Enables (SLGON) or disables (SLGOFF) logging of system time changes.
SLG_MANDV	SLGON	Enables (SLGON) or disables (SLGOFF) logging of mandatory access requests.

Parameter	Default value	Description
SLG_MAXSIZE	8192000	Size of security log to retire. When SLG_FILE is larger than this, it is moved to a file prefixed with SLG_FPREFIX.
SLG_MKDIRV	SLGON	Enables (SLGON) or disables (SLGOFF) logging of all mkdir violations.
SLG_NETWV	SLGON	Enables (SLGON) or disables (SLGOFF) logging of network access violations.
SLG_PATH_TRACK	SLGON	Enables (SLGON) or disables (SLGOFF) tracking of all path names on accesses.
SLG_PHYSIO_ERR	SLGOFF	Enables (SLGON) or disables (SLGOFF) logging of all physical I/O errors.
SLG_REMOVEV	SLGON	Enables (SLGON) or disables (SLGOFF) logging of all rm violations.
SLG_RMDIRV	SLGON	Enables (SLGON) or disables (SLGOFF) logging of all rmdir violations.
SLG_STATE	SLGOFF	Enables (SLGON) or disables (SLGOFF) security log.
SLG_SUID_RQ	SLGON	Enables (SLGON) or disables (SLGOFF) logging of setuid system calls.
SLG_SULOG	SLGON	Enables (SLGON) or disables (SLGOFF) logging of all su(1) command attempts.
SLG_T_PROC	SLGON	Enables (SLGON) or disables (SLGOFF) logging of trusted process activity.
SLG_USER	SLGOFF	Enables (SLGON) or disables (SLGOFF) logging of user names and passwords for failed login attempts.
SYSTEM_ADMIN_CONSOLE	"/dev/console"	Default console for MLS administration. The value of this parameter must be /dev/console.
SYSVCOMPS	0	Bit map that defines valid system compartments. By default, system compartments are turned off.

Table 9 contains the parameter in config.h that relates to TCP/IP. See the *UNICOS Networking Facilities Administrator's Guide*, Cray Research publication SG–2304, for more information on setting this parameter.

Table 9. TCP/IP kernel parameter in config.h

Parameter	Default value	Description
TCP_NMBSPACE	3800	Controls the number of MBUFs (TCP/IP managed memory buffers). For information about appropriate values, see *UNICOS Networking Facilities Administrator's Guide*, Cray Research publication SG–2304. This pool of buffers is allocated at boot time in module /usr/src/uts/tcp/kern/uipc_mbuf and cannot be increased while the system is running. The ICMS places this value in both the config.h file and the CSL parameter file. The value in the CSL parameter file overrides the value in config.h.

Table 10 shows the NFS kernel parameters in config.h.

Table 10. NFS kernel parameters in config.h

Parameter	Default value	Description
NFS3_ASYNC_MAX	64	The amount of data (in 4096–byte blocks) that will be written per file asynchronously. After this value is exceeded, all writes will be synchronous. (2000 * 4096 = 8,192,000 bytes.)
NFS3_ASYNC_TIME	120	The amount of time (in seconds) that data will be held in the NFS asynchronous write cache on the client.
NFS_DUPTIMEOUT	3	Time interval in seconds during which duplicates will not be replayed.
NFS_MAXDATA	32768	Maximum user data read or written by way of NFS.
NFS_MAXDUPREQS	1200	Size of the duplicate request cache. This value must be large enough so that the entry will still be there when the first retry comes in.

Parameter	Default value	Description
`NFS_NUM_RNODES`	`256`	Number of NFS nodes that may be read from (known as *rnodes*) at any one time in the kernel. Each active NFS file or directory requires an rnode. Files may have multiple requests outstanding at any one time. If the number of active NFS files exceeds the `NFS_NUM_RNODES` value, the rnodes are shared among active files.
`NFS_PRINTINTER`	`0`	Time interval between two messages on the console window when the server is not responding.
`CNFS_STATIC_CLIENTS`	`8`	Number of CNFS static client handles.
`NFS_STATIC_CLIENTS`	`8`	Number of NFS static client handles.
`CNFS_TEMP_CLIENTS`	`8`	Number of CNFS temporary client handles.
`NFS_TEMP_CLIENTS`	`8`	Number of NFS temporary client handles.

Some of the NFS kernel parameters have CSL boot-time equivalents, as shown in Table 11. These boot-time equivalents override the values that were set in the `config.h` file when the kernel was built. The CSL tags and their values will override the built-in kernel values, and change the performance of the UNICOS operating system.

Table 11. CSL boot-time equivalents for NFS kernel parameters

CSL tag	config.h
nfs3_async_max	NFS3_ASYNC_MAX
nfs3_async_time	NFS3_ASYNC_TIME
nfs_duptimeout	NFS_DUPTIMEOUT
nfs_maxdata	NFS_MAXDATA
nfs_maxdupreqs	NFS_MAXDUPREQS
nfs_num_rnodes	NFS_NUM_RNODES
nfs_printinter	NFS_PRINTINTER
cnfs_static_clients	CNFS_STATIC_CLIENTS
nfs_static_clients	NFS_STATIC_CLIENTS
cnfs_temp_clients	CNFS_TEMP_CLIENTS
nfs_temp_clients	NFS_TEMP_CLIENTS