

# Linux System Call Table

The following table lists the **system calls** for the **Linux 2.2** kernel. It could also be thought of as an API for the interface between user space and kernel space. My motivation for making this table was to make programming in assembly language easier when using only **system calls** and not the C library (for more information on this topic, go to <http://www.linuxassembly.org>). On the left are the numbers of the **system calls**. This number will be put in register `%eax`. On the right of the table are the types of values to be put into the remaining registers before calling the software interrupt 'int 0x80'. After each syscall, an integer is returned in `%eax`.

For convenience, links go from the "Name" column to the man page for most of the **system calls**. Links to the kernel source file where each **system call** is located are linked to in the column labelled "Source". (You can also [download](#) a version of this page which has links directly to the source that is installed on your **system**.) Links to definitions are provided for the parameters that are typedefs or structs.

<code>%eax</code>	Name	Source	<code>%ebx</code>	<code>%ecx</code>	<code>%edx</code>	<code>%esx</code>	<code>%edi</code>
1	<a href="#">sys_exit</a>	<a href="#">kernel/exit.c</a>	int	-	-	-	-
2	<a href="#">sys_fork</a>	<a href="#">arch/i386/kernel/process.c</a>	<a href="#">struct pt_regs</a>	-	-	-	-
3	<a href="#">sys_read</a>	<a href="#">fs/read_write.c</a>	unsigned int	char *	<a href="#">size_t</a>	-	-
4	<a href="#">sys_write</a>	<a href="#">fs/read_write.c</a>	unsigned int	const char *	<a href="#">size_t</a>	-	-
5	<a href="#">sys_open</a>	<a href="#">fs/open.c</a>	const char *	int	int	-	-
6	<a href="#">sys_close</a>	<a href="#">fs/open.c</a>	unsigned int	-	-	-	-
7	<a href="#">sys_waitpid</a>	<a href="#">kernel/exit.c</a>	pid_t	unsigned int *	int	-	-
8	<a href="#">sys_creat</a>	<a href="#">fs/open.c</a>	const char *	int	-	-	-
9	<a href="#">sys_link</a>	<a href="#">fs/namei.c</a>	const char *	const char *	-	-	-
10	<a href="#">sys_unlink</a>	<a href="#">fs/namei.c</a>	const char *	-	-	-	-
11	<a href="#">sys_execve</a>	<a href="#">arch/i386/kernel/process.c</a>	<a href="#">struct pt_regs</a>	-	-	-	-
12	<a href="#">sys_chdir</a>	<a href="#">fs/open.c</a>	const char *	-	-	-	-
13	<a href="#">sys_time</a>	<a href="#">kernel/time.c</a>	int *	-	-	-	-
14	<a href="#">sys_mknod</a>	<a href="#">fs/namei.c</a>	const char *	int	<a href="#">dev_t</a>	-	-
15	<a href="#">sys_chmod</a>	<a href="#">fs/open.c</a>	const char *	<a href="#">mode_t</a>	-	-	-
16	<a href="#">sys_lchown</a>	<a href="#">fs/open.c</a>	const char *	<a href="#">uid_t</a>	<a href="#">gid_t</a>	-	-
18	<a href="#">sys_stat</a>	<a href="#">fs/stat.c</a>	char *	<a href="#">struct _old_kernel_stat *</a>	-	-	-
19	<a href="#">sys_lseek</a>	<a href="#">fs/read_write.c</a>	unsigned int	<a href="#">off_t</a>	unsigned int	-	-
20	<a href="#">sys_getpid</a>	<a href="#">kernel/sched.c</a>	-	-	-	-	-
21	<a href="#">sys_mount</a>	<a href="#">fs/super.c</a>	char *	char *	char *	-	-
22	<a href="#">sys_oldumount</a>	<a href="#">fs/super.c</a>	char *	-	-	-	-
23	<a href="#">sys_setuid</a>	<a href="#">kernel/sys.c</a>	<a href="#">uid_t</a>	-	-	-	-
24	<a href="#">sys_getuid</a>	<a href="#">kernel/sched.c</a>	-	-	-	-	-
25	<a href="#">sys_stime</a>	<a href="#">kernel/time.c</a>	int *	-	-	-	-
26	<a href="#">sys_ptrace</a>	<a href="#">arch/i386/kernel/ptrace.c</a>	long	long	long	long	-
27	<a href="#">sys_alarm</a>	<a href="#">kernel/sched.c</a>	unsigned int	-	-	-	-
28	<a href="#">sys_fstat</a>	<a href="#">fs/stat.c</a>	unsigned int	<a href="#">struct _old_kernel_stat *</a>	-	-	-
29	<a href="#">sys_pause</a>	<a href="#">arch/i386/kernel/sys_i386.c</a>	-	-	-	-	-
30	<a href="#">sys_utime</a>	<a href="#">fs/open.c</a>	char *	<a href="#">struct utimbuf *</a>	-	-	-
33	<a href="#">sys_access</a>	<a href="#">fs/open.c</a>	const char *	int	-	-	-
34	<a href="#">sys_nice</a>	<a href="#">kernel/sched.c</a>	int	-	-	-	-
36	<a href="#">sys_sync</a>	<a href="#">fs/buffer.c</a>	-	-	-	-	-
37	<a href="#">sys_kill</a>	<a href="#">kernel/signal.c</a>	int	int	-	-	-
38	<a href="#">sys_rename</a>	<a href="#">fs/namei.c</a>	const char *	const char *	-	-	-
39	<a href="#">sys_mkdir</a>	<a href="#">fs/namei.c</a>	const char *	int	-	-	-
40	<a href="#">sys_rmdir</a>	<a href="#">fs/namei.c</a>	const char *	-	-	-	-
41	<a href="#">sys_dup</a>	<a href="#">fs/fcntl.c</a>	unsigned int	-	-	-	-
42	<a href="#">sys_pipe</a>	<a href="#">arch/i386/kernel/sys_i386.c</a>	unsigned long *	-	-	-	-
43	<a href="#">sys_times</a>	<a href="#">kernel/sys.c</a>	<a href="#">struct tms *</a>	-	-	-	-
45	<a href="#">sys_brk</a>	<a href="#">mm/mmap.c</a>	unsigned long	-	-	-	-

46	<a href="#">sys_setgid</a>	<a href="#">kernel/sys.c</a>	<a href="#">gid_t</a>	-	-	-	-
47	<a href="#">sys_getgid</a>	<a href="#">kernel/sched.c</a>	-	-	-	-	-
48	<a href="#">sys_signal</a>	<a href="#">kernel/signal.c</a>	int	<a href="#">_sighandler_t</a>	-	-	-
49	<a href="#">sys_geteuid</a>	<a href="#">kernel/sched.c</a>	-	-	-	-	-
50	<a href="#">sys_getegid</a>	<a href="#">kernel/sched.c</a>	-	-	-	-	-
51	<a href="#">sys_acct</a>	<a href="#">kernel/acct.c</a>	const char *	-	-	-	-
52	<a href="#">sys_umount</a>	<a href="#">fs/super.c</a>	char *	int	-	-	-
54	<a href="#">sys_ioctl</a>	<a href="#">fs/ioctl.c</a>	unsigned int	unsigned int	unsigned long	-	-
55	<a href="#">sys_fcntl</a>	<a href="#">fs/fcntl.c</a>	unsigned int	unsigned int	unsigned long	-	-
57	<a href="#">sys_setpgid</a>	<a href="#">kernel/sys.c</a>	<a href="#">pid_t</a>	<a href="#">pid_t</a>	-	-	-
59	<a href="#">sys_olduname</a>	<a href="#">arch/i386/kernel/sys_i386.c</a>	<a href="#">struct oldold_utsname *</a>	-	-	-	-
60	<a href="#">sys_umask</a>	<a href="#">kernel/sys.c</a>	int	-	-	-	-
61	<a href="#">sys_chroot</a>	<a href="#">fs/open.c</a>	const char *	-	-	-	-
62	<a href="#">sys_ustat</a>	<a href="#">fs/super.c</a>	<a href="#">dev_t</a>	<a href="#">struct ustat *</a>	-	-	-
63	<a href="#">sys_dup2</a>	<a href="#">fs/fcntl.c</a>	unsigned int	unsigned int	-	-	-
64	<a href="#">sys_getppid</a>	<a href="#">kernel/sched.c</a>	-	-	-	-	-
65	<a href="#">sys_getpgrp</a>	<a href="#">kernel/sys.c</a>	-	-	-	-	-
66	<a href="#">sys_setsid</a>	<a href="#">kernel/sys.c</a>	-	-	-	-	-
67	<a href="#">sys_sigaction</a>	<a href="#">arch/i386/kernel/signal.c</a>	int	const <a href="#">struct old_sigaction *</a>	<a href="#">struct old_sigaction *</a>	-	-
68	<a href="#">sys_sgetmask</a>	<a href="#">kernel/signal.c</a>	-	-	-	-	-
69	<a href="#">sys_ssetmask</a>	<a href="#">kernel/signal.c</a>	int	-	-	-	-
70	<a href="#">sys_setreuid</a>	<a href="#">kernel/sys.c</a>	<a href="#">uid_t</a>	<a href="#">uid_t</a>	-	-	-
71	<a href="#">sys_setregid</a>	<a href="#">kernel/sys.c</a>	<a href="#">gid_t</a>	<a href="#">gid_t</a>	-	-	-
72	<a href="#">sys_sigsuspend</a>	<a href="#">arch/i386/kernel/signal.c</a>	int	int	<a href="#">old_sigset_t</a>	-	-
73	<a href="#">sys_sigpending</a>	<a href="#">kernel/signal.c</a>	<a href="#">old_sigset_t *</a>	-	-	-	-
74	<a href="#">sys_sethostname</a>	<a href="#">kernel/sys.c</a>	char *	int	-	-	-
75	<a href="#">sys_setrlimit</a>	<a href="#">kernel/sys.c</a>	unsigned int	<a href="#">struct rlimit *</a>	-	-	-
76	<a href="#">sys_getrlimit</a>	<a href="#">kernel/sys.c</a>	unsigned int	<a href="#">struct rlimit *</a>	-	-	-
77	<a href="#">sys_getrusage</a>	<a href="#">kernel/sys.c</a>	int	<a href="#">struct rusage *</a>	-	-	-
78	<a href="#">sys_gettimeofday</a>	<a href="#">kernel/time.c</a>	<a href="#">struct timeval *</a>	<a href="#">struct timezone *</a>	-	-	-
79	<a href="#">sys_settimeofday</a>	<a href="#">kernel/time.c</a>	<a href="#">struct timeval *</a>	<a href="#">struct timezone *</a>	-	-	-
80	<a href="#">sys_getgroups</a>	<a href="#">kernel/sys.c</a>	int	<a href="#">gid_t *</a>	-	-	-
81	<a href="#">sys_setgroups</a>	<a href="#">kernel/sys.c</a>	int	<a href="#">gid_t *</a>	-	-	-
82	old_select	<a href="#">arch/i386/kernel/sys_i386.c</a>	<a href="#">struct sel_arg_struct *</a>	-	-	-	-
83	<a href="#">sys_symlink</a>	<a href="#">fs/namei.c</a>	const char *	const char *	-	-	-
84	<a href="#">sys_lstat</a>	<a href="#">fs/stat.c</a>	char *	<a href="#">struct _old_kernel_stat *</a>	-	-	-
85	<a href="#">sys_readlink</a>	<a href="#">fs/stat.c</a>	const char *	char *	int	-	-
86	<a href="#">sys_uselib</a>	<a href="#">fs/exec.c</a>	const char *	-	-	-	-
87	<a href="#">sys_swapon</a>	<a href="#">mm/swapfile.c</a>	const char *	int	-	-	-
88	<a href="#">sys_reboot</a>	<a href="#">kernel/sys.c</a>	int	int	int	void *	-
89	old_readdir	<a href="#">fs/readdir.c</a>	unsigned int	void *	unsigned int	-	-
90	old_mmap	<a href="#">arch/i386/kernel/sys_i386.c</a>	<a href="#">struct mmap_arg_struct *</a>	-	-	-	-
91	<a href="#">sys_munmap</a>	<a href="#">mm/mmap.c</a>	unsigned long	<a href="#">size_t</a>	-	-	-
92	<a href="#">sys_truncate</a>	<a href="#">fs/open.c</a>	const char *	unsigned long	-	-	-
93	<a href="#">sys_ftruncate</a>	<a href="#">fs/open.c</a>	unsigned int	unsigned long	-	-	-

94	<a href="#">sys_fchmod</a>	<a href="#">fs/open.c</a>	unsigned int	<a href="#">mode_t</a>	-	-	-
95	<a href="#">sys_fchown</a>	<a href="#">fs/open.c</a>	unsigned int	<a href="#">uid_t</a>	<a href="#">gid_t</a>	-	-
96	<a href="#">sys_getpriority</a>	<a href="#">kernel/sys.c</a>	int	int	-	-	-
97	<a href="#">sys_setpriority</a>	<a href="#">kernel/sys.c</a>	int	int	int	-	-
99	<a href="#">sys_statfs</a>	<a href="#">fs/open.c</a>	const char *	<a href="#">struct statfs *</a>	-	-	-
100	<a href="#">sys_fstatfs</a>	<a href="#">fs/open.c</a>	unsigned int	<a href="#">struct statfs *</a>	-	-	-
101	<a href="#">sys_ioperm</a>	<a href="#">arch/i386/kernel/ioport.c</a>	unsigned long	unsigned long	int	-	-
102	<a href="#">sys_socketcall</a>	<a href="#">net/socket.c</a>	int	unsigned long *	-	-	-
103	<a href="#">sys_syslog</a>	<a href="#">kernel/printk.c</a>	int	char *	int	-	-
104	<a href="#">sys_setitimer</a>	<a href="#">kernel/itimer.c</a>	int	<a href="#">struct itimerval *</a>	<a href="#">struct itimerval *</a>	-	-
105	<a href="#">sys_getitimer</a>	<a href="#">kernel/itimer.c</a>	int	<a href="#">struct itimerval *</a>	-	-	-
106	sys_newstat	<a href="#">fs/stat.c</a>	char *	<a href="#">struct stat *</a>	-	-	-
107	sys_newlstat	<a href="#">fs/stat.c</a>	char *	<a href="#">struct stat *</a>	-	-	-
108	sys_newfstat	<a href="#">fs/stat.c</a>	unsigned int	<a href="#">struct stat *</a>	-	-	-
109	<a href="#">sys_uname</a>	<a href="#">arch/i386/kernel/sys_i386.c</a>	<a href="#">struct old_utsname *</a>	-	-	-	-
110	<a href="#">sys_iopl</a>	<a href="#">arch/i386/kernel/ioport.c</a>	unsigned long	-	-	-	-
111	<a href="#">sys_vhangup</a>	<a href="#">fs/open.c</a>	-	-	-	-	-
112	<a href="#">sys_idle</a>	<a href="#">arch/i386/kernel/process.c</a>	-	-	-	-	-
113	<a href="#">sys_vm86old</a>	<a href="#">arch/i386/kernel/vm86.c</a>	unsigned long	<a href="#">struct vm86plus_struct *</a>	-	-	-
114	<a href="#">sys_wait4</a>	<a href="#">kernel/exit.c</a>	<a href="#">pid_t</a>	unsigned long *	int options	<a href="#">struct rusage *</a>	-
115	<a href="#">sys_swapoff</a>	<a href="#">mm/swapfile.c</a>	const char *	-	-	-	-
116	<a href="#">sys_sysinfo</a>	<a href="#">kernel/info.c</a>	<a href="#">struct sysinfo *</a>	-	-	-	-
117	sys_ipc(*Note)	<a href="#">arch/i386/kernel/sys_i386.c</a>	<a href="#">uint</a>	int	int	int	void *
118	<a href="#">sys_fsync</a>	<a href="#">fs/buffer.c</a>	unsigned int	-	-	-	-
119	<a href="#">sys_sigreturn</a>	<a href="#">arch/i386/kernel/signal.c</a>	unsigned long	-	-	-	-
120	<a href="#">sys_clone</a>	<a href="#">arch/i386/kernel/process.c</a>	<a href="#">struct pt_regs</a>	-	-	-	-
121	<a href="#">sys_setdomainname</a>	<a href="#">kernel/sys.c</a>	char *	int	-	-	-
122	sys_newuname	<a href="#">kernel/sys.c</a>	<a href="#">struct new_utsname *</a>	-	-	-	-
123	<a href="#">sys_modify_ldt</a>	<a href="#">arch/i386/kernel/ldt.c</a>	int	void *	unsigned long	-	-
124	<a href="#">sys_adjtimex</a>	<a href="#">kernel/time.c</a>	<a href="#">struct timex *</a>	-	-	-	-
125	<a href="#">sys_mprotect</a>	<a href="#">mm/mprotect.c</a>	unsigned long	<a href="#">size_t</a>	unsigned long	-	-
126	<a href="#">sys_sigprocmask</a>	<a href="#">kernel/signal.c</a>	int	<a href="#">old_sigset_t *</a>	<a href="#">old_sigset_t *</a>	-	-
127	<a href="#">sys_create_module</a>	<a href="#">kernel/module.c</a>	const char *	<a href="#">size_t</a>	-	-	-
128	<a href="#">sys_init_module</a>	<a href="#">kernel/module.c</a>	const char *	<a href="#">struct module *</a>	-	-	-
129	<a href="#">sys_delete_module</a>	<a href="#">kernel/module.c</a>	const char *	-	-	-	-
130	<a href="#">sys_get_kernel_syms</a>	<a href="#">kernel/module.c</a>	<a href="#">struct kernel_sym *</a>	-	-	-	-
131	<a href="#">sys_quotactl</a>	<a href="#">fs/dquot.c</a>	int	const char *	int	<a href="#">caddr_t</a>	-
132	<a href="#">sys_getpgid</a>	<a href="#">kernel/sys.c</a>	<a href="#">pid_t</a>	-	-	-	-
133	<a href="#">sys_fchdir</a>	<a href="#">fs/open.c</a>	unsigned int	-	-	-	-
134	<a href="#">sys_bdflush</a>	<a href="#">fs/buffer.c</a>	int	long	-	-	-
135	<a href="#">sys_sysfs</a>	<a href="#">fs/super.c</a>	int	unsigned long	unsigned long	-	-
136	<a href="#">sys_personality</a>	<a href="#">kernel/exec_domain.c</a>	unsigned long	-	-	-	-
138	<a href="#">sys_setsuid</a>	<a href="#">kernel/sys.c</a>	<a href="#">uid_t</a>	-	-	-	-
139	<a href="#">sys_setfsuid</a>	<a href="#">kernel/sys.c</a>	<a href="#">gid_t</a>	-	-	-	-
					unsigned		unsigned

140	<a href="#">sys_llseek</a>	<a href="#">fs/read_write.c</a>	unsigned int	unsigned long	long	<a href="#">loff_t *</a>	int
141	<a href="#">sys_getdents</a>	<a href="#">fs/readdir.c</a>	unsigned int	void *	unsigned int	-	-
142	<a href="#">sys_select</a>	<a href="#">fs/select.c</a>	int	<a href="#">fd_set *</a>	<a href="#">fd_set *</a>	<a href="#">fd_set *</a>	<a href="#">struct timeval *</a>
143	<a href="#">sys_flock</a>	<a href="#">fs/locks.c</a>	unsigned int	unsigned int	-	-	-
144	<a href="#">sys_msync</a>	<a href="#">mm/filemap.c</a>	unsigned long	<a href="#">size_t</a>	int	-	-
145	<a href="#">sys_readv</a>	<a href="#">fs/read_write.c</a>	unsigned long	<a href="#">const struct iovec *</a>	unsigned long	-	-
146	<a href="#">sys_writev</a>	<a href="#">fs/read_write.c</a>	unsigned long	<a href="#">const struct iovec *</a>	unsigned long	-	-
147	<a href="#">sys_getsid</a>	<a href="#">kernel/sys.c</a>	<a href="#">pid_t</a>	-	-	-	-
148	<a href="#">sys_fdatasync</a>	<a href="#">fs/buffer.c</a>	unsigned int	-	-	-	-
149	<a href="#">sys_sysctl</a>	<a href="#">kernel/sysctl.c</a>	<a href="#">struct _sysctl_args *</a>	-	-	-	-
150	<a href="#">sys_mlock</a>	<a href="#">mm/mlock.c</a>	unsigned long	<a href="#">size_t</a>	-	-	-
151	<a href="#">sys_munlock</a>	<a href="#">mm/mlock.c</a>	unsigned long	<a href="#">size_t</a>	-	-	-
152	<a href="#">sys_mlockall</a>	<a href="#">mm/mlock.c</a>	int	-	-	-	-
153	<a href="#">sys_munlockall</a>	<a href="#">mm/mlock.c</a>	-	-	-	-	-
154	<a href="#">sys_sched_setparam</a>	<a href="#">kernel/sched.c</a>	<a href="#">pid_t</a>	<a href="#">struct sched_param *</a>	-	-	-
155	<a href="#">sys_sched_getparam</a>	<a href="#">kernel/sched.c</a>	<a href="#">pid_t</a>	<a href="#">struct sched_param *</a>	-	-	-
156	<a href="#">sys_sched_setscheduler</a>	<a href="#">kernel/sched.c</a>	<a href="#">pid_t</a>	int	<a href="#">struct sched_param *</a>	-	-
157	<a href="#">sys_sched_getscheduler</a>	<a href="#">kernel/sched.c</a>	<a href="#">pid_t</a>	-	-	-	-
158	<a href="#">sys_sched_yield</a>	<a href="#">kernel/sched.c</a>	-	-	-	-	-
159	<a href="#">sys_sched_get_priority_max</a>	<a href="#">kernel/sched.c</a>	int	-	-	-	-
160	<a href="#">sys_sched_get_priority_min</a>	<a href="#">kernel/sched.c</a>	int	-	-	-	-
161	<a href="#">sys_sched_rr_get_interval</a>	<a href="#">kernel/sched.c</a>	<a href="#">pid_t</a>	<a href="#">struct timespec *</a>	-	-	-
162	<a href="#">sys_nanosleep</a>	<a href="#">kernel/sched.c</a>	<a href="#">struct timespec *</a>	<a href="#">struct timespec *</a>	-	-	-
163	<a href="#">sys_mremap</a>	<a href="#">mm/mremap.c</a>	unsigned long	unsigned long	unsigned long	unsigned long	-
164	<a href="#">sys_setresuid</a>	<a href="#">kernel/sys.c</a>	<a href="#">uid_t</a>	<a href="#">uid_t</a>	<a href="#">uid_t</a>	-	-
165	<a href="#">sys_getresuid</a>	<a href="#">kernel/sys.c</a>	<a href="#">uid_t *</a>	<a href="#">uid_t *</a>	<a href="#">uid_t *</a>	-	-
166	<a href="#">sys_vm86</a>	<a href="#">arch/i386/kernel/vm86.c</a>	<a href="#">struct vm86_struct *</a>	-	-	-	-
167	<a href="#">sys_query_module</a>	<a href="#">kernel/module.c</a>	const char *	int	char *	<a href="#">size_t</a>	<a href="#">size_t *</a>
168	<a href="#">sys_poll</a>	<a href="#">fs/select.c</a>	<a href="#">struct pollfd *</a>	unsigned int	long	-	-
169	<a href="#">sys_nfservctl</a>	<a href="#">fs/filesystems.c</a>	int	void *	void *	-	-
170	<a href="#">sys_setresgid</a>	<a href="#">kernel/sys.c</a>	<a href="#">gid_t</a>	<a href="#">gid_t</a>	<a href="#">gid_t</a>	-	-
171	<a href="#">sys_getresgid</a>	<a href="#">kernel/sys.c</a>	<a href="#">gid_t *</a>	<a href="#">gid_t *</a>	<a href="#">gid_t *</a>	-	-
172	<a href="#">sys_prctl</a>	<a href="#">kernel/sys.c</a>	int	unsigned long	unsigned long	unsigned long	unsigned long
173	<a href="#">sys_rt_sigreturn</a>	<a href="#">arch/i386/kernel/signal.c</a>	unsigned long	-	-	-	-
174	<a href="#">sys_rt_sigaction</a>	<a href="#">kernel/signal.c</a>	int	<a href="#">const struct sigaction *</a>	<a href="#">struct sigaction *</a>	<a href="#">size_t</a>	-
175	<a href="#">sys_rt_sigprocmask</a>	<a href="#">kernel/signal.c</a>	int	<a href="#">sigset_t *</a>	<a href="#">sigset_t *</a>	<a href="#">size_t</a>	-
176	<a href="#">sys_rt_sigpending</a>	<a href="#">kernel/signal.c</a>	<a href="#">sigset_t *</a>	<a href="#">size_t</a>	-	-	-
177	<a href="#">sys_rt_sigtimedwait</a>	<a href="#">kernel/signal.c</a>	<a href="#">const sigset_t *</a>	<a href="#">siginfo_t *</a>	<a href="#">const struct timespec *</a>	<a href="#">size_t</a>	-
178	<a href="#">sys_rt_sigqueueinfo</a>	<a href="#">kernel/signal.c</a>	int	int	<a href="#">siginfo_t *</a>	-	-
179	<a href="#">sys_rt_sigsuspend</a>	<a href="#">arch/i386/kernel/signal.c</a>	<a href="#">sigset_t *</a>	<a href="#">size_t</a>	-	-	-
180	<a href="#">sys_pread</a>	<a href="#">fs/read_write.c</a>	unsigned int	char *	<a href="#">size_t</a>	<a href="#">loff_t</a>	-
181	<a href="#">sys_pwrite</a>	<a href="#">fs/read_write.c</a>	unsigned int	const char *	<a href="#">size_t</a>	<a href="#">loff_t</a>	-

182	<a href="#">sys_chown</a>	<a href="#">fs/open.c</a>	const char *	<a href="#">uid_t</a>	<a href="#">gid_t</a>	-	-
183	<a href="#">sys_getcwd</a>	<a href="#">fs/dcache.c</a>	char *	unsigned long	-	-	-
184	<a href="#">sys_capget</a>	<a href="#">kernel/capability.c</a>	<a href="#">cap_user_header_t</a>	<a href="#">cap_user_data_t</a>	-	-	-
185	<a href="#">sys_capset</a>	<a href="#">kernel/capability.c</a>	<a href="#">cap_user_header_t</a>	<a href="#">const cap_user_data_t</a>	-	-	-
186	<a href="#">sys_sigaltstack</a>	<a href="#">arch/i386/kernel/signal.c</a>	<a href="#">const stack_t *</a>	<a href="#">stack_t *</a>	-	-	-
187	<a href="#">sys_sendfile</a>	<a href="#">mm/filemap.c</a>	int	int	<a href="#">off_t *</a>	<a href="#">size_t</a>	-
190	<a href="#">sys_vfork</a>	<a href="#">arch/i386/kernel/process.c</a>	<a href="#">struct pt_regs</a>	-	-	-	-

**Note for sys\_ipc (117):** this syscall takes six arguments, so it can't fit into the five registers %ebx - %edi; the last parameter (not shown) is of type 'long'. This syscall requires a special call method where a pointer is put in %ebx which points to an array containing the six arguments.

## System Call Numbers

For the numbers of the syscalls, look in [arch/i386/kernel/entry.S](#) for **sys\_call\_table**. The syscall numbers are offsets into that table. Several spots in the table are occupied by the syscall **sys\_ni\_syscall**. This is a placeholder that either replaces an obsolete syscall or reserves a spot for future syscalls.

Incidentally, the **system calls** are called from the function **system\_call** in the same file; in particular, they are called with the assembly instruction 'call \*SYMBOL\_NAME(sys\_call\_table)(,%eax,4)'. The part '\*SYMBOL\_NAME(sys\_call\_table)' just gets replaced by a symbol name in **sys\_call\_table**. **SYMBOL\_NAME** is a macro defined in [include/linux/linkage.h](#), and it just replaces itself with its argument.

## Typedefs

Here are the typedef declarations in the prototypes above:

<b>atomic_t</b>	<a href="#">include/asm/atomic.h</a> : #ifdef __SMP__ typedef struct { volatile int counter; } atomic_t; #else typedef struct { int counter; } atomic_t; #endif
<b>caddr_t</b>	<a href="#">include/asm/posix_types.h</a> :typedef char * __kernel_caddr_t; <a href="#">include/linux/types.h</a> :typedef __kernel_caddr_t caddr_t;
<b>cap_user_header_t</b>	<a href="#">include/linux/capability.h</a> : typedef struct __user_cap_header_struct { __u32 version; int pid; } *cap_user_header_t;
<b>cap_user_data_t</b>	<a href="#">include/linux/capability.h</a> : typedef struct __user_cap_data_struct { __u32 effective; __u32 permitted; __u32 inheritable; } *cap_user_data_t;
<b>clock_t</b>	<a href="#">include/asm/posix_types.h</a> :typedef long __kernel_clock_t; <a href="#">include/linux/types.h</a> :typedef __kernel_clock_t clock_t;
<b>dev_t</b>	<a href="#">include/asm/posix_types.h</a> :typedef unsigned short __kernel_dev_t; <a href="#">include/linux/types.h</a> :typedef __kernel_dev_t dev_t;
<b>fd_set</b>	<a href="#">include/linux/posix_types.h</a> #define __FD_SETSIZE 1024 #define __NFDBITS (8 * sizeof(unsigned long)) #define __FDSET_LONGS (__FD_SETSIZE / __NFDBITS) (==> __FDSET_LONGS == 32)  typedef struct { unsigned long fds_bits [ __FDSET_LONGS]; } __kernel_fd_set; <a href="#">include/linux/types.h</a> :typedef __kernel_fd_set fd_set;
<b>gid_t</b>	<a href="#">include/asm/posix_types.h</a> :typedef unsigned short __kernel_gid_t; <a href="#">include/linux/types.h</a> :typedef __kernel_gid_t gid_t;
<b>__kernel_daddr_t</b>	<a href="#">include/asm/posix_types.h</a> :typedef int __kernel_daddr_t;

<b>__kernel_fsid_t</b>	<a href="#">include/asm/posix_types.h</a> : typedef struct { int __val[2]; } __kernel_fsid_t;
<b>__kernel_ino_t</b>	<a href="#">include/asm/posix_types.h</a> :typedef unsigned long __kernel_ino_t;
<b>__kernel_size_t</b>	<a href="#">include/asm/posix_types.h</a> :typedef unsigned int __kernel_size_t;
<b>loff_t</b>	<a href="#">include/asm/posix_types.h</a> :typedef long long __kernel_loff_t; <a href="#">include/linux/types.h</a> :typedef __kernel_loff_t loff_t;
<b>mode_t</b>	<a href="#">include/asm/posix_types.h</a> :typedef unsigned short __kernel_mode_t; <a href="#">include/linux/types.h</a> :typedef __kernel_mode_t mode_t;
<b>off_t</b>	<a href="#">include/asm/posix_types.h</a> :typedef long __kernel_off_t; <a href="#">include/linux/types.h</a> :typedef __kernel_off_t off_t;
<b>old_sigset_t</b>	<a href="#">include/asm/signal.h</a> :typedef unsigned long old_sigset_t;
<b>pid_t</b>	<a href="#">include/asm/posix_types.h</a> :typedef int __kernel_pid_t; <a href="#">include/linux/types.h</a> :typedef __kernel_pid_t pid_t;
<b>__sighandler_t</b>	<a href="#">include/asm/signal.h</a> :typedef void (*__sighandler_t)(int);
<b>siginfo_t</b>	<a href="#">include/asm/siginfo.h</a> : #define SI_MAX_SIZE 128 #define SI_PAD_SIZE ((SI_MAX_SIZE/sizeof(int)) - 3) (==> SI_PAD_SIZE == 29)  typedef struct siginfo { int si_signo; int si_errno; int si_code;  union { int _pad[SI_PAD_SIZE];  /* kill() */ struct { <a href="#">pid_t</a> _pid; /* sender's pid */ <a href="#">uid_t</a> _uid; /* sender's uid */ } _kill;  /* POSIX.1b timers */ struct { unsigned int _timer1; unsigned int _timer2; } _timer;  /* POSIX.1b signals */ struct { <a href="#">pid_t</a> _pid; /* sender's pid */ <a href="#">uid_t</a> _uid; /* sender's uid */ <a href="#">sigval_t</a> _sigval; } _rt;  /* SIGCHLD */ struct { <a href="#">pid_t</a> _pid; /* which child */ <a href="#">uid_t</a> _uid; /* sender's uid */ int _status; /* exit code */ <a href="#">clock_t</a> _utime; <a href="#">clock_t</a> _stime; } _sigchld;  /* SIGILL, SIGFPE, SIGSEGV, SIGBUS */ struct { void *_addr; /* faulting insn/memory ref. */ } _sigfault;  /* SIGPOLL */ struct { int _band; /* POLL_IN, POLL_OUT, POLL_MSG */ int _fd; } _sigpoll; } _sifields; };

	} siginfo_t;
<b>sigset_t</b>	<a href="#">include/asm/signal.h</a> :typedef unsigned long sigset_t;
<b>size_t</b>	<a href="#">include/asm/posix_types.h</a> :typedef unsigned int __kernel_size_t; <a href="#">include/linux/types.h</a> :typedef __kernel_size_t size_t;
<b>ssize_t</b>	<a href="#">include/asm/posix_types.h</a> :typedef int __kernel_ssize_t; <a href="#">include/linux/types.h</a> :typedef __kernel_ssize_t ssize_t;
<b>stack_t</b>	<a href="#">include/asm/signal.h</a> : typedef struct sigaltstack { void *ss_sp; int ss_flags; <a href="#">size_t</a> ss_size; } stack_t;
<b>suseconds_t</b>	<a href="#">include/asm/posix_types.h</a> :typedef long __kernel_suseconds_t; <a href="#">include/linux/types.h</a> :typedef __kernel_suseconds_t suseconds_t;
<b>time_t</b>	<a href="#">include/asm/posix_types.h</a> :typedef long __kernel_time_t; <a href="#">include/linux/types.h</a> :typedef __kernel_time_t time_t;
<b>uid_t</b>	<a href="#">include/asm/posix_types.h</a> :typedef unsigned short __kernel_uid_t; <a href="#">include/linux/types.h</a> :typedef __kernel_uid_t uid_t;
<b>uint</b>	<a href="#">include/linux/types.h</a> :typedef unsigned int uint;
<b>__u32</b>	<a href="#">include/asm/types.h</a> :typedef unsigned int __u32;

## Structs

Here are the struct declarations for the table at the top:

<b>exception_table_entry</b>	<a href="#">include/linux/module.h</a> : struct exception_table_entry { unsigned long insn, fixup; };
<b>iovec</b>	<a href="#">include/linux/uio.h</a> : struct iovec { void *iov_base; <a href="#">__kernel_size_t</a> iov_len; };
<b>itimerval</b>	<a href="#">include/linux/time.h</a> : struct itimerval { <a href="#">struct timeval</a> it_interval; /* timer interval */ <a href="#">struct timeval</a> it_value; /* current value */ };
<b>kernel_sym</b>	<a href="#">include/linux/module.h</a> : struct kernel_sym { unsigned long value; char name[60]; };
<b>mmap_arg_struct</b>	<a href="#">arch/i386/kernel/sys_i386.c</a> : struct mmap_arg_struct { unsigned long addr; unsigned long len; unsigned long prot; unsigned long flags; unsigned long fd; unsigned long offset; };
<b>module</b>	<a href="#">include/linux/module.h</a> : struct module { unsigned long size_of_struct; /* sizeof(module) */ <a href="#">struct module</a> *next; const char *name; unsigned long size; union { <a href="#">atomic_t</a> usecount; long pad; } uc; unsigned long flags; /* AUTOCLEAN et al */ };



	<pre> unsigned nsyms; unsigned ndeps;  struct module_symbol *syms; struct module_ref *deps; struct module_ref *refs; int (*init)(void); void (*cleanup)(void); const struct exception_table_entry *ex_table_start; const struct exception_table_entry *ex_table_end; /* Members past this point are extensions to the basic module support and are optional. Use mod_opt_member() to examine them. */ const struct module_persist *persist_start; const struct module_persist *persist_end; int (*can_unload)(void); }; </pre>
<b>module_persist</b>	<pre> include/linux/module.h: struct module_persist; /* yes, it's empty */ </pre>
<b>module_ref</b>	<pre> include/linux/module.h: struct module_ref {     struct module *dep; /* "parent" pointer */     struct module *ref; /* "child" pointer */     struct module_ref *next_ref; }; </pre>
<b>module_symbol</b>	<pre> include/linux/module.h: struct module_symbol {     unsigned long value;     const char *name; }; </pre>
<b>new_utsname</b>	<pre> include/linux/utsname.h: struct new_utsname {     char sysname[65];     char nodename[65];     char release[65];     char version[65];     char machine[65];     char domainname[65]; }; </pre>
<b>__old_kernel_stat</b>	<pre> include/asm/stat.h: struct __old_kernel_stat {     unsigned short st_dev;     unsigned short st_ino;     unsigned short st_mode;     unsigned short st_nlink;     unsigned short st_uid;     unsigned short st_gid;     unsigned short st_rdev;     unsigned long st_size;     unsigned long st_atime;     unsigned long st_mtime;     unsigned long st_ctime; }; </pre>
<b>oldold_utsname</b>	<pre> include/linux/utsname.h: struct oldold_utsname {     char sysname[9];     char nodename[9];     char release[9];     char version[9];     char machine[9]; }; </pre>
<b>old_sigaction</b>	<pre> include/asm/signal.h: struct old_sigaction {     __sighandler_t sa_handler;     old_sigset_t sa_mask;     unsigned long sa_flags;     void (*sa_restorer)(void); }; </pre>



<b>old_utsname</b>	<a href="#">include/linux/utsname.h</a> : <pre>struct old_utsname {     char sysname[65];     char nodename[65];     char release[65];     char version[65];     char machine[65]; };</pre>
<b>pollfd</b>	<a href="#">include/asm/poll.h</a> : <pre>struct pollfd {     int fd;     short events;     short revents; };</pre>
<b>pt_regs</b>	<a href="#">include/asm/ptrace.h</a> : <pre>struct pt_regs {     long ebx;     long ecx;     long edx;     long esi;     long edi;     long ebp;     long eax;     int xds;     int xes;     long orig_eax;     long eip;     int xcs;     long eflags;     long esp;     int xss; };</pre>
<b>revector_struct</b>	<a href="#">include/asm/vm86.h</a> : <pre>struct revector_struct {     unsigned long __map[8]; };</pre>
<b>rlimit</b>	<a href="#">include/linux/resource.h</a> : <pre>struct rlimit {     long rlim_cur;     long rlim_max; };</pre>
<b>rusage</b>	<a href="#">include/linux/resource.h</a> : <pre>struct rusage {     <a href="#">struct timeval</a> ru_utime; /* user time used */     <a href="#">struct timeval</a> ru_stime; /* <b>system</b> time used */     long ru_maxrss; /* maximum resident set size */     long ru_ixrss; /* integral shared memory size */     long ru_idrss; /* integral unshared data size */     long ru_isrss; /* integral unshared stack size */     long ru_minflt; /* page reclaims */     long ru_majflt; /* page faults */     long ru_nswap; /* swaps */     long ru_inblock; /* block input operations */     long ru_oublock; /* block output operations */     long ru_msgsnd; /* messages sent */     long ru_msgrcv; /* messages received */     long ru_nsignals; /* signals received */     long ru_nvcsw; /* voluntary context switches */     long ru_nivcsw; /* involuntary " */ };</pre>
<b>sched_param</b>	<a href="#">include/linux/sched.h</a> : <pre>struct sched_param {     int sched_priority; };</pre>
<b>sel_arg_struct</b>	<a href="#">arch/i386/kernel/sys_i386.c</a> : <pre>struct sel_arg_struct {     unsigned long n;     <a href="#">fd_set</a> *inp, *outp, *exp;     <a href="#">struct timeval</a> *tvp; };</pre>

	};
<b>sigaction</b>	<a href="#">include/asm/signal.h:</a> struct sigaction { <a href="#">__sighandler_t</a> sa_handler; unsigned long sa_flags; void (*sa_restorer)(void); <a href="#">sigset_t</a> sa_mask; /* mask last for extensibility */ };
<b>stat</b>	<a href="#">include/asm/stat.h:</a> struct stat { unsigned short st_dev; unsigned short __pad1; unsigned long st_ino; unsigned short st_mode; unsigned short st_nlink; unsigned short st_uid; unsigned short st_gid; unsigned short st_rdev; unsigned short __pad2; unsigned long st_size; unsigned long st_blksize; unsigned long st_blocks; unsigned long st_atime; unsigned long __unused1; unsigned long st_mtime; unsigned long __unused2; unsigned long st_ctime; unsigned long __unused3; unsigned long __unused4; unsigned long __unused5; };
<b>statfs</b>	<a href="#">include/asm/statfs.h:</a> struct statfs { long f_type; long f_bsize; long f_blocks; long f_bfree; long f_bavail; long f_files; long f_ffree; <a href="#">__kernel_fsid_t</a> f_fsid; long f_namelen; long f_spare[6]; };
<b>__sysctl_args</b>	<a href="#">include/linux/sysctl.h</a> struct __sysctl_args { int *name; int nlen; void *oldval; <a href="#">size_t</a> *oldlenp; void *newval; <a href="#">size_t</a> newlen; unsigned long __unused[4]; };
<b>sysinfo</b>	<a href="#">include/linux/kernel.h:</a> struct sysinfo { long uptime; /* Seconds since boot */ unsigned long loads[3]; /* 1, 5, and 15 minute load averages */ unsigned long totalram; /* Total usable main memory size */ unsigned long freeram; /* Available memory size */ unsigned long sharedram; /* Amount of shared memory */ unsigned long bufferram; /* Memory used by buffers */ unsigned long totalswap; /* Total swap space size */ unsigned long freeswap; /* swap space still available */ unsigned short procs; /* Number of current processes */ char _f[22]; /* Pads structure to 64 bytes */ };
<b>timex</b>	<a href="#">include/linux/timex.h:</a> struct timex { unsigned int modes; /* mode selector */ };

	<pre> long offset; /* time offset (usec) */ long freq; /* frequency offset (scaled ppm) */ long maxerror; /* maximum error (usec) */ long esterror; /* estimated error (usec) */ int status; /* clock command/status */ long constant; /* pll time constant */ long precision; /* clock precision (usec) (read only) */ long tolerance; /* clock frequency tolerance (ppm) * (read only) */ <a href="#">struct timeval</a> time; /* (read only) */ long tick; /* (modified) usecs between clock ticks */ long ppsfreq; /* pps frequency (scaled ppm) (ro) */ long jitter; /* pps jitter (us) (ro) */ int shift; /* interval duration (s) (shift) (ro) */ long stabil; /* pps stability (scaled ppm) (ro) */ long jitcnt; /* jitter limit exceeded (ro) */ long calcnt; /* calibration intervals (ro) */ long errcnt; /* calibration errors (ro) */ long stbcnt; /* stability limit exceeded (ro) */  int :32; int :32; int :32; int :32; int :32; int :32; int :32; int :32; int :32; int :32; int :32; int :32; }; </pre>
<b>timespec</b>	<pre> <a href="#">include/linux/time.h</a>: struct timespec {     <a href="#">time_t</a> tv_sec; /* seconds */     long tv_nsec; /* nanoseconds */ }; </pre>
<b>timeval</b>	<pre> <a href="#">include/linux/time.h</a>: struct timeval {     <a href="#">time_t</a> tv_sec; /* seconds */     <a href="#">suseconds_t</a> tv_usec; /* microseconds */ }; </pre>
<b>timezone</b>	<pre> <a href="#">include/linux/time.h</a>: struct timezone {     int tz_minuteswest; /* minutes west of Greenwich */     int tz_dsttime; /* type of dst correction */ }; </pre>
<b>tms</b>	<pre> <a href="#">include/linux/times.h</a> struct tms {     <a href="#">clock_t</a> tms_utime;     <a href="#">clock_t</a> tms_stime;     <a href="#">clock_t</a> tms_cutime;     <a href="#">clock_t</a> tms_cstime; }; </pre>
<b>ustat</b>	<pre> <a href="#">include/linux/types.h</a>: struct ustat {     <a href="#">__kernel_daddr_t</a> f_tfree;     <a href="#">__kernel_ino_t</a> f_tinode;     char f_fname[6];     char f_fpack[6]; }; </pre>
<b>utimbuf</b>	<pre> <a href="#">include/linux/utime.h</a>: struct utimbuf {     <a href="#">time_t</a> actime;     <a href="#">time_t</a> modtime; }; </pre>
<b>vm86plus_info_struct</b>	<pre> <a href="#">include/asm/vm86.h</a>: struct vm86plus_info_struct {     unsigned long force_return_for_pic:1;     unsigned long vm86dbg_active:1;     unsigned long vm86dbg_TFpendig:1;     unsigned long unused:28;     unsigned long is_vm86pus:1;     unsigned char vm86dbg_intxxtab[32]; }; </pre>

<b>vm86plus_struct</b>	<a href="#">include/asm/vm86.h</a> : <pre> struct vm86plus_struct {     <a href="#">struct vm86_regs</a> regs;     unsigned long flags;     unsigned long screen_bitmap;     unsigned long cpu_type;     <a href="#">struct revector</a> <a href="#">struct</a> int_revector;     <a href="#">struct revector</a> <a href="#">struct</a> int21_revector;     <a href="#">struct vm86plus_info_struct</a> vm86plus; }; </pre>
<b>vm86_regs</b>	<a href="#">include/asm/vm86.h</a> : <pre> struct vm86_regs { /* normal regs, with special meaning for the segment descriptors.. */     long ebx;     long ecx;     long edx;     long esi;     long edi;     long ebp;     long eax;     long __null_ds;     long __null_es;     long __null_fs;     long __null_gs;     long orig_eax;     long eip;     unsigned short cs, __csh;     long eflags;     long esp;     unsigned short ss, __ssh; /* these are specific to v86 mode: */     unsigned short es, __esh;     unsigned short ds, __dsh;     unsigned short fs, __fsh;     unsigned short gs, __gsh; }; </pre>
<b>vm86_struct</b>	<a href="#">include/asm/vm86.h</a> : <pre> struct vm86_struct {     <a href="#">struct vm86_regs</a> regs;     unsigned long flags;     unsigned long screen_bitmap;     unsigned long cpu_type;     <a href="#">struct revector</a> <a href="#">struct</a> int_revector;     <a href="#">struct revector</a> <a href="#">struct</a> int21_revector; }; </pre>