(source)

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, the kernel source file where each system call is located is linked to in the column labelled "Source". In order to use the hyperlinks, you must first copy this page to your own machine because the links take you directly to the source code on your system. You must have the kernel source installed (or linked from) under '/usr/src/linux' for this to work.

%eax	Name	Source	%ebx	
1	sys_exit	kernel/exit.c	int	-
2	sys_fork	arch/i386/kernel/process.c	struct pt_regs	-
3	sys_read	fs/read_write.c	unsigned int	char
4	sys_write	fs/read_write.c	unsigned int	cons
5	sys_open	fs/open.c	const char *	int
6	sys_close	fs/open.c	unsigned int	-
7	sys_waitpid	kernel/exit.c	pid_t	unsi
8	sys_creat	fs/open.c	const char *	int
9	sys_link	fs/namei.c	const char *	cons
10	sys_unlink	fs/namei.c	const char *	-
11	sys_execve	arch/i386/kernel/process.c	struct pt_regs	-
12	sys_chdir	fs/open.c	const char *	-
13	sys_time	kernel/time.c	int *	-
14	sys_mknod	fs/namei.c	const char *	int
15	sys_chmod	fs/open.c	const char *	mod
16	sys_lchown	fs/open.c	const char *	uid_
18	sys_stat	fs/stat.c	char *	stru __old *
19	sys_lseek	fs/read_write.c	unsigned int	off_t
20	sys_getpid	kernel/sched.c	-	-
21	sys_mount	fs/super.c	char *	char
22	sys_oldumount	fs/super.c	char *	-
23	sys_setuid	kernel/sys.c	uid_t	-
24	sys_getuid	kernel/sched.c	-	-
25	sys_stime	kernel/time.c	int *	-
26	sys_ptrace	arch/i386/kernel/ptrace.c	long	long

27	sys_alarm	kernel/sched.c	unsigned int	-
28	sys_fstat	fs/stat.c	unsigned int	stru __old *
29	sys_pause	arch/i386/kernel/sys_i386.c	-	-
30	sys_utime	fs/open.c	char *	stru
33	sys_access	fs/open.c	const char *	int
34	sys_nice	kernel/sched.c	int	-
36	sys_sync	fs/buffer.c	-	-
37	sys_kill	kernel/signal.c	int	int
38	sys_rename	fs/namei.c	const char *	cons
39	sys_mkdir	fs/namei.c	const char *	int
40	sys_rmdir	fs/namei.c	const char *	-
41	sys_dup	fs/fcntl.c	unsigned int	-
42	sys_pipe	arch/i386/kernel/sys_i386.c	unsigned long *	-
43	sys_times	kernel/sys.c	struct tms *	-
45	sys_brk	mm/mmap.c	unsigned long	-
46	sys_setgid	kernel/sys.c	gid_t	-
47	sys_getgid	kernel/sched.c	-	-
48	sys_signal	kernel/signal.c	int	__sig
49	sys_geteuid	kernel/sched.c	-	-
50	sys_getegid	kernel/sched.c	-	-
51	sys_acct	kernel/acct.c	const char *	-
52	sys_umount	fs/super.c	char *	int
54	sys_ioctl	fs/ioctl.c	unsigned int	unsi
55	sys_fcntl	fs/fcntl.c	unsigned int	unsi
57	sys_setpgid	kernel/sys.c	pid_t	pid_
59	sys_olduname	arch/i386/kernel/sys_i386.c	struct oldold_utsname *	-
60	sys_umask	kernel/sys.c	int	-
61	sys_chroot	fs/open.c	const char *	-
62	sys_ustat	fs/super.c	dev_t	stru
63	sys_dup2	fs/fcntl.c	unsigned int	unsi
64	sys_getppid	kernel/sched.c	-	-
65	sys_getpgrp	kernel/sys.c	-	-
66	sys_setsid	kernel/sys.c	-	-
67	sys_sigaction	arch/i386/kernel/signal.c	int	cons old_s
68	sys_sgetmask	kernel/signal.c	-	-
69	sys_ssetmask	kernel/signal.c	int	-
70	sys_setreuid	kernel/sys.c	uid_t	uid_
71	sys_setregid	kernel/sys.c	gid_t	gid_

72	sys_sigsuspend	arch/i386/kernel/signal.c	int	int
73	sys_sigpending	kernel/signal.c	old_sigset_t *	-
74	sys_sethostname	kernel/sys.c	char *	int
75	sys_setrlimit	kernel/sys.c	unsigned int	stru
76	sys_getrlimit	kernel/sys.c	unsigned int	stru
77	sys_getrusage	kernel/sys.c	int	stru
78	sys_gettimeofday	kernel/time.c	struct timeval *	stru *
79	sys_settimeofday	kernel/time.c	struct timeval *	stru *
80	sys_getgroups	kernel/sys.c	int	gid_
81	sys_setgroups	kernel/sys.c	int	gid_
82	old_select	arch/i386/kernel/sys_i386.c	struct sel_arg_struct *	-
83	sys_symlink	fs/namei.c	const char *	cons
84	sys_lstat	fs/stat.c	char *	stru __old *
85	sys_readlink	fs/stat.c	const char *	char
86	sys_uselib	fs/exec.c	const char *	-
87	sys_swapon	mm/swapfile.c	const char *	int
88	sys_reboot	kernel/sys.c	int	int
89	old_readdir	fs/readdir.c	unsigned int	void
90	old_mmap	arch/i386/kernel/sys_i386.c	struct mmap_arg_struct *	-
91	sys_munmap	mm/mmap.c	unsigned long	size_
92	sys_truncate	fs/open.c	const char *	unsi
93	sys_ftruncate	fs/open.c	unsigned int	unsi
94	sys_fchmod	fs/open.c	unsigned int	mod
95	sys_fchown	fs/open.c	unsigned int	uid_
96	sys_getpriority	kernel/sys.c	int	int
97	sys_setpriority	kernel/sys.c	int	int
99	sys_statfs	fs/open.c	const char *	stru
100	sys_fstatfs	fs/open.c	unsigned int	stru
101	sys_ioperm	arch/i386/kernel/ioport.c	unsigned long	unsi
102	sys_socketcall	net/socket.c	int	unsi
103	sys_syslog	kernel/printk.c	int	char
104	sys_setitimer	kernel/itimer.c	int	stru
105	sys_getitimer	kernel/itimer.c	int	stru
106	sys_newstat	fs/stat.c	char *	stru
107	sys_newlstat	fs/stat.c	char *	stru
108	sys_newfstat	fs/stat.c	unsigned int	stru

109	sys_uname	arch/i386/kernel/sys_i386.c	struct old_utsname *	-
110	sys_iopl	arch/i386/kernel/ioport.c	unsigned long	-
111	sys_vhangup	fs/open.c	-	-
112	sys_idle	arch/i386/kernel/process.c	-	-
113	sys_vm86old	arch/i386/kernel/vm86.c	unsigned long	stru vm8 *
114	sys_wait4	kernel/exit.c	pid_t	unsi
115	sys_swapoff	mm/swapfile.c	const char *	-
116	sys_sysinfo	kernel/info.c	struct sysinfo *	-
117	sys_ipc (*Note)	arch/i386/kernel/sys_i386.c	uint	int
118	sys_fsync	fs/buffer.c	unsigned int	-
119	sys_sigreturn	arch/i386/kernel/signal.c	unsigned long	-
120	sys_clone	arch/i386/kernel/process.c	struct pt_regs	-
121	sys_setdomainname	kernel/sys.c	char *	int
122	sys_newuname	kernel/sys.c	struct new_utsname *	-
123	sys_modify_ldt	arch/i386/kernel/ldt.c	int	void
124	sys_adjtimex	kernel/time.c	struct timex *	-
125	sys_mprotect	mm/mprotect.c	unsigned long	size_
126	sys_sigprocmask	kernel/signal.c	int	old_
127	sys_create_module	kernel/module.c	const char *	size_
128	sys_init_module	kernel/module.c	const char *	stru
129	sys_delete_module	kernel/module.c	const char *	-
130	sys_get_kernel_syms	kernel/module.c	struct kernel_sym *	-
131	sys_quotactl	fs/dquot.c	int	cons
132	sys_getpgid	kernel/sys.c	pid_t	-
133	sys_fchdir	fs/open.c	unsigned int	-
134	sys_bdflush	fs/buffer.c	int	long
135	sys_sysfs	fs/super.c	int	unsi
136	sys_personality	kernel/exec_domain.c	unsigned long	-
138	sys_setfsuid	kernel/sys.c	uid_t	-
139	sys_setfsgid	kernel/sys.c	gid_t	-
140	sys_llseek	fs/read_write.c	unsigned int	unsi
141	sys_getdents	fs/readdir.c	unsigned int	void
142	sys_select	fs/select.c	int	fd_s
143	sys_flock	fs/locks.c	unsigned int	unsi

144	sys_msync	mm/filemap.c	unsigned long	size_
145	sys_readv	fs/read_write.c	unsigned long	cons iove
146	sys_writev	fs/read_write.c	unsigned long	cons iove
147	sys_getsid	kernel/sys.c	pid_t	-
148	sys_fdatasync	fs/buffer.c	unsigned int	-
149	sys_sysctl	kernel/sysctl.c	struct __sysctl_args *	-
150	sys_mlock	mm/mlock.c	unsigned long	size_
151	sys_munlock	mm/mlock.c	unsigned long	size_
152	sys_mlockall	mm/mlock.c	int	-
153	sys_munlockall	mm/mlock.c	-	-
154	sys_sched_setparam	kernel/sched.c	pid_t	stru sche
155	sys_sched_getparam	kernel/sched.c	pid_t	stru sche
156	sys_sched_setscheduler	kernel/sched.c	pid_t	int
157	sys_sched_getscheduler	kernel/sched.c	pid_t	-
158	sys_sched_yield	kernel/sched.c	-	-
159	sys_sched_get_priority_max	kernel/sched.c	int	-
160	sys_sched_get_priority_min	kernel/sched.c	int	-
161	sys_sched_rr_get_interval	kernel/sched.c	pid_t	stru
162	sys_nanosleep	kernel/sched.c	struct timespec *	stru
163	sys_mremap	mm/mremap.c	unsigned long	unsi
164	sys_setresuid	kernel/sys.c	uid_t	uid_
165	sys_getresuid	kernel/sys.c	uid_t *	uid_
166	sys_vm86	arch/i386/kernel/vm86.c	struct vm86_struct *	-
167	sys_query_module	kernel/module.c	const char *	int
168	sys_poll	fs/select.c	struct pollfd *	unsi
169	sys_nfsservctl	fs/filesystems.c	int	void
170	sys_setresgid	kernel/sys.c	gid_t	gid_
171	sys_getresgid	kernel/sys.c	gid_t *	gid_
172	sys_prctl	kernel/sys.c	int	unsi
173	sys_rt_sigreturn	arch/i386/kernel/signal.c	unsigned long	-
174	sys_rt_sigaction	kernel/signal.c	int	cons siga
175	sys_rt_sigprocmask	kernel/signal.c	int	sigse
176	sys_rt_sigpending	kernel/signal.c	sigset_t *	size_
177	sys_rt_sigtimedwait	kernel/signal.c	const sigset_t *	sigin

178	sys_rt_sigqueueinfo	kernel/signal.c	int	int
179	sys_rt_sigsuspend	arch/i386/kernel/signal.c	sigset_t *	size_
180	sys_pread	fs/read_write.c	unsigned int	char
181	sys_pwrite	fs/read_write.c	unsigned int	cons
182	sys_chown	fs/open.c	const char *	uid_
183	sys_getcwd	fs/dcache.c	char *	unsi
184	sys_capget	kernel/capability.c	cap_user_header_t	cap_
185	sys_capset	kernel/capability.c	cap_user_header_t	cons cap_
186	sys_sigaltstack	arch/i386/kernel/signal.c	const stack_t *	stac
187	sys_sendfile	mm/filemap.c	int	int
190	sys_vfork	arch/i386/kernel/process.c	struct pt_regs	-

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.

I will now explain exactly where in the kernel source that I got the information in the table above. I do this because 1) changes in the source are bound to happen, 2) you might be curious, or 3) I might've made an error.

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:

atomic_t	include/asm/atomic.h: #ifdef __SMP__ typedef struct { volatile int counter; } atomic_t; #else typedef struct { int counter; } atomic_t; #endif
caddr_t	include/asm/posix_types.h:typedef char * __kernel_caddr_t; include/linux/types.h:typedef __kernel_caddr_t caddr_t;
cap_user_header_t	include/linux/capability.h: typedef struct __user_cap_header_struct { __u32 version; int pid;

	} *cap_user_header_t;
cap_user_data_t	include/linux/capability.h: typedef struct __user_cap_data_struct { __u32 effective; __u32 permitted; __u32 inheritable; } *cap_user_data_t;
clock_t	include/asm/posix_types.h:typedef long __kernel_clock_t; include/linux/types.h:typedef __kernel_clock_t clock_t;
dev_t	include/asm/posix_types.h:typedef unsigned short __kernel_dev_t; include/linux/types.h:typedef __kernel_dev_t dev_t;
fdset	include/linux/posix_types.h #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; include/linux/types.h:typedef __kernel_fd_set fd_set;
gid_t	include/asm/posix_types.h:typedef unsigned short __kernel_gid_t; include/linux/types.h:typedef __kernel_gid_t gid_t;
__kernel_daddr_t	include/asm/posix_types.h:typedef int __kernel_daddr_t;
__kernel_fsid_t	include/asm/posix_types.h: typedef struct { int __val[2]; } __kernel_fsid_t;
__kernel_ino_t	include/asm/posix_types.h:typedef unsigned long __kernel_ino_t;
__kernel_size_t	include/asm/posix_types.h:typedef unsigned int __kernel_size_t;
loff_t	include/asm/posix_types.h:typedef long long __kernel_loff_t; include/linux/types.h:typedef __kernel_loff_t loff_t;
mode_t	include/asm/posix_types.h:typedef unsigned short __kernel_mode_t; include/linux/types.h:typedef __kernel_mode_t mode_t;
off_t	include/asm/posix_types.h:typedef long __kernel_off_t; include/linux/types.h:typedef __kernel_off_t off_t;
old_sigset_t	include/asm/signal.h:typedef unsigned long old_sigset_t;
pid_t	include/asm/posix_types.h:typedef int __kernel_pid_t; include/linux/types.h:typedef __kernel_pid_t pid_t;
__sighandler_t	include/asm/signal.h:typedef void (*__sighandler_t)(int);
siginfo_t	include/asm/siginfo.h: #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 {
	pid_t _pid; /* sender's pid */
	uid_t _uid; /* sender's uid */
	} _kill;
	
	/* POSIX.1b timers */
	struct {
	unsigned int _timer1;
	unsigned int _timer2;
	} _timer;
	
	/* POSIX.1b signals */
	struct {
	pid_t _pid; /* sender's pid */
	uid_t _uid; /* sender's uid */
	sigval_t _sigval;
	} _rt;
	
	/* SIGCHLD */
	struct {
	pid_t _pid; /* which child */
	uid_t _uid; /* sender's uid */
	int _status; /* exit code */
	clock_t _utime;
	clock_t _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;

sigset_t	include/asm/signal.h:typedef unsigned long sigset_t;
size_t	include/asm/posix_types.h:typedef unsigned int __kernel_size_t; include/linux/types.h:typedef __kernel_size_t size_t;
ssize_t	include/asm/posix_types.h:typedef int __kernel_ssize_t; include/linux/types.h:typedef __kernel_ssize_t ssize_t;
stack_t	include/asm/signal.h: typedef struct sigaltstack { void *ss_sp; int ss_flags; size_t ss_size; } stack_t;
suseconds_t	include/asm/posix_types.h:typedef long __kernel_suseconds_t; include/linux/types.h:typedef __kernel_suseconds_t suseconds_t;

time_t	include/asm/posix_types.h:typedef long __kernel_time_t; include/linux/types.h:typedef __kernel_time_t time_t;
uid_t	include/asm/posix_types.h:typedef unsigned short __kernel_uid_t; include/linux/types.h:typedef __kernel_uid_t uid_t;
uint	include/linux/types.h:typedef unsigned int uint;
__u32	include/asm/types.h:typedef unsigned int __u32;

Struct Declarations

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

exception_table_entry	include/linux/module.h:
	struct exception_table_entry {
	  unsigned long insn, fixup;
	};

iovec	include/linux/uio.h:
	struct iovec {
	  void *iov_base;
	  __kernel_size_t iov_len; };

itimerval	include/linux/time.h:
	struct itimerval {
	  struct timeval it_interval; /* timer interval */
	  struct timeval it_value; /* current value */
	};

kernel_sym	include/linux/module.h:
	struct kernel_sym {
	  unsigned long value;
	  char name[60];
	};

mmap_arg_struct	arch/i386/kernel/sys_i386.c:
	struct mmap_arg_struct {
	  unsigned long addr;
	  unsigned long len;
	  unsigned long prot;
	  unsigned long flags;
	  unsigned long fd;
	  unsigned long offset;
	};

module	include/linux/module.h:
	struct module {
	  unsigned long size_of_struct; /* sizeof(module) */
	  struct module *next;
	  const char *name;
	  unsigned long size;
	  union {
	  atomic_t usecount;
	  long pad;
	  } uc;
	  unsigned long flags; /* AUTOCLEAN et al */
	  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); };
module_persist	include/linux/module.h: struct module_persist; /* yes, it's empty */

module_ref	include/linux/module.h:
	struct module_ref {
	struct module *dep; /* "parent" pointer */
	struct module *ref; /* "child" pointer */
	struct module_ref *next_ref;
	};

module_symbol	include/linux/module.h:
	struct module_symbol {
	unsigned long value;
	const char *name;
	};

new_utsname	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];
	};

__old_kernel_stat	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;
	};

oldold_utsname	include/linux/utsname.h: struct oldold_utsname { char sysname[9]; char nodename[9]; char release[9]; char version[9];

	char machine[9]; };

old_sigaction	include/asm/signal.h:
	struct old_sigaction {
	__sighandler_t sa_handler;
	old_sigset_t sa_mask;
	unsigned long sa_flags;
	void (*sa_restorer)(void);
	};

old_utsname	include/linux/utsname.h:
	struct old_utsname {
	char sysname[65];
	char nodename[65];
	char release[65];
	char version[65];
	char machine[65];
	};

pollfd	include/asm/poll.h:
	struct pollfd {
	int fd;
	short events;
	short revents;
	};

pt_regs	include/asm/ptrace.h:
	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;
	};

revectored_struct	include/asm/vm86.h:
	struct revectored_struct {
	unsigned long __map[8];
	};

rlimit	include/linux/resource.h:
	struct rlimit {
	long rlim_cur;
	long rlim_max;
	};

rusage	include/linux/resource.h: struct rusage { struct timeval ru_utime; /* user time used */ struct timeval ru_stime; /* system 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 '' */ };

sched_param	include/linux/sched.h:
	struct sched_param {
	int sched_priority;
	};

sel_arg_struct	arch/i386/kernel/sys_i386.c:
	struct sel_arg_struct {
	unsigned long n;
	fd_set *inp, *outp, *exp;
	struct timeval *tvp;
	};

sigaction	include/asm/signal.h:
	struct sigaction {
	__sighandler_t sa_handler;
	unsigned long sa_flags;
	void (*sa_restorer)(void);
	sigset_t sa_mask; /* mask last for extensibility */
	};

stat	include/asm/stat.h: 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;
statfs	include/asm/statfs.h: struct statfs { long f_type; long f_bsize; long f_blocks;

	long f_bfree; long f_bavail; long f_files; long f_ffree; __kernel_fsid_t f_fsid; long f_namelen; long f_spare[6]; };

__sysctl_args	include/linux/sysctl.h
	struct __sysctl_args {
	int *name;
	int nlen;
	void *oldval;
	size_t *oldlenp;
	void *newval;
	size_t newlen;
	unsigned long __unused[4];
	};

sysinfo	include/linux/kernel.h:
	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 */
	};

timex	include/linux/timex.h: struct timex { unsigned int modes; /* mode selector */ 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) */ struct timeval 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; };

timespec	include/linux/time.h:
	struct timespec {
	time_t tv_sec; /* seconds */
	long tv_nsec; /* nanoseconds */
	};

timeval	include/linux/time.h:
	struct timeval {
	time_t tv_sec; /* seconds */
	suseconds_t tv_usec; /* microseconds */
	};

timezone	include/linux/time.h:
	struct timezone {
	int tz_minuteswest; /* minutes west of Greenwich */
	int tz_dsttime; /* type of dst correction */
	};

tms	include/linux/times.h
	struct tms {
	clock_t tms_utime;
	clock_t tms_stime;
	clock_t tms_cutime;
	clock_t tms_cstime;
	};

ustat	include/linux/types.h:
	struct ustat {
	__kernel_daddr_t f_tfree;
	__kernel_ino_t f_tinode;
	char f_fname[6];
	char f_fpack[6];
	};

utimbuf	include/linux/utime.h:
	struct utimbuf {
	time_t actime;
	time_t modtime;
	};

vm86plus_info_struct	include/asm/vm86.h:
	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];
	};

vm86plus_struct	include/asm/vm86.h: struct vm86plus_struct { struct vm86_regs regs; unsigned long flags; unsigned long screen_bitmap; unsigned long cpu_type; struct revectored_struct int_revectored; struct revectored_struct int21_revectored; struct vm86plus_info_struct vm86plus;

	};

vm86_regs	include/asm/vm86.h:
	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;
	};

vm86_struct	include/asm/vm86.h:
	struct vm86_struct {
	struct vm86_regs regs;
	unsigned long flags;
	unsigned long screen_bitmap;
	unsigned long cpu_type;
	struct revectored_struct int_revectored;
	struct revectored_struct int21_revectored;
	};