x86 (32-bit) System Call Reference - Top 100 Most Common

Overview

This document contains the 100 most commonly used system call codes for the x86 32-bit architecture (i386). System calls are invoked using interrupt 0x80 (int 0x80) with the syscall number in EAX and arguments in EBX, ECX, EDX, ESI, EDI, and EBP registers.

Register Usage: -

Register	Function
EAX	System call number
EBX	1st argument
ECX	2nd argument
EDX	3rd argument
ESI	4th argument
EDI	5th argument
EBP	6th argument

Syscall Directives: -

Code	Name	Arguments	Description
1	exit	int error_code	Terminate the calling process
2	fork	-	Create a child process
3	read	unsigned int fd, char *buf, size_t count	Read from file descriptor

Code	Name	Arguments	Description
4	write	unsigned int fd, const char *buf, size_t count	Write to file descriptor
5	open	const char *filename, int flags, umode_t mode	Open file
6	close	unsigned int fd	Close file descriptor
7	waitpid	pid_t pid, int *stat_addr, int options	Wait for process to change state
8	creat	const char *pathname, umode_t mode	Create file
9	link	const char *oldname, const char* newname	Create hard link
10	unlink	const char *pathname	Delete file
11	execve	const char *filename, const char* const argv[], const char *const envp[]	Execute program
12	chdir	const char *filename	Change working directory
13	time	time_t *tloc	Get time in seconds
14	mknod	const char *filename, umode_t mode, unsigned dev	Create device special file
15	chmod	const char *filename, umode_t mode	Change file permissions
16	lchown	const char *filename, uid_t user, gid_t group	Change file ownership
19	lseek	unsigned int fd, off_t offset, unsigned int whence	Reposition file offset
20	getpid	-	Get process ID
21	mount	char *dev_name, char* dir_name, char *type,	Mount filesystem

Code	Name	Arguments	Description
		unsigned long flags, void data	
22	umount	char *name	Unmount filesystem
23	setuid	uid_t uid	Set user ID
24	getuid	-	Get user ID
26	ptrace	long request, long pid, unsigned long addr, unsigned long data	Process trace
27	alarm	unsigned int seconds	Set alarm clock
29	pause	-	Suspend process until signal
30	utime	char *filename, struct utimbuf* times	Change file timestamps
33	access	const char *filename, int mode	Check file permissions
34	nice	int increment	Change process priority
36	sync	-	Synchronize cached writes
37	kill	pid_t pid, int sig	Send signal to process
38	rename	const char *oldname, const char* newname	Rename file
39	mkdir	const char *pathname, umode_t mode	Create directory
40	rmdir	const char *pathname	Remove directory
41	dup	unsigned int fildes	Duplicate file descriptor
42	pipe	int *fildes	Create pipe
43	times	struct tms *tbuf	Get process times
45	brk	unsigned long brk	Change data segment size
46	setgid	gid_t gid	Set group ID
47	getgid	-	Get group ID
48	signal	int sig, __sighandler_t handler	Install signal handler
49	geteuid	-	Get effective user ID
50	getegid	-	Get effective group ID

Code	Name	Arguments	Description
54	ioctl	unsigned int fd, unsigned int cmd, unsigned long arg	Device control
55	fcntl	unsigned int fd, unsigned int cmd, unsigned long arg	File control
57	setpgid	pid_t pid, pid_t pgid	Set process group
60	umask	int mask	Set file creation mask
61	chroot	const char *filename	Change root directory
63	dup2	unsigned int oldfd, unsigned int newfd	Duplicate file descriptor
64	getppid	-	Get parent process ID
65	getpgrp	-	Get process group
66	setsid	-	Create session
67	sigaction	int sig, const struct old_sigaction *act, struct old_sigaction* oact	Examine/change signal action
70	setreuid	uid_t ruid, uid_t euid	Set real/effective user ID
71	setregid	gid_t rgid, gid_t egid	Set real/effective group ID
72	sigsuspend	old_sigset_t mask	Wait for signal
73	sigpending	old_sigset_t *set	Examine pending signals
75	setrlimit	unsigned int resource, struct rlimit *rlim	Set resource limit
76	getrlimit	unsigned int resource, struct rlimit *rlim	Get resource limit
77	getrusage	int who, struct rusage *ru	Get resource usage
78	gettimeofday	struct timeval *tv, struct timezone* tz	Get time of day
79	settimeofday	struct timeval *tv, struct timezone* tz	Set time of day

Code	Name	Arguments	Description
80	getgroups	int gidsetsize, gid_t *grouplist	Get supplementary group IDs
81	setgroups	int gidsetsize, gid_t *grouplist	Set supplementary group IDs
82	select	int n, fd_set *inp, fd_set* outp, fd_set *exp, struct timeval* tvp	Synchronous I/O multiplexing
83	symlink	const char *oldname, const char* newname	Create symbolic link
85	readlink	const char *path, char* buf, int bufsiz	Read symbolic link
87	swapon	const char *specialfile, int swap_flags	Start swapping
88	reboot	int magic1, int magic2, unsigned int cmd, void *arg	Reboot system
90	mmap	unsigned long addr, unsigned long len, unsigned long prot, unsigned long flags, unsigned long fd, unsigned long off	Map files/devices into memory
91	munmap	unsigned long addr, size_t len	Unmap files/devices from memory
92	truncate	const char *path, long length	Truncate file
93	ftruncate	unsigned int fd, unsigned long length	Truncate file
94	fchmod	unsigned int fd, umode_t mode	Change file permissions
95	fchown	unsigned int fd, uid_t user, gid_t group	Change file ownership
96	getpriority	int which, int who	Get process priority
97	setpriority	int which, int who, int niceval	Set process priority
99	statfs	const char *pathname, struct statfs* buf	Get filesystem statistics

Code	Name	Arguments	Description
100	fstatfs	unsigned int fd, struct statfs *buf	Get filesystem statistics
102	socketcall	int call, unsigned long *args	Socket system calls
103	syslog	int type, char *buf, int len	Read/clear kernel message ring buffer
104	setitimer	int which, struct itimerval *value, struct itimerval* ovalue	Set interval timer
105	getitimer	int which, struct itimerval *value	Get interval timer
106	stat	const char *filename, struct __old_kernel_stat *statbuf	Get file status
107	lstat	const char *filename, struct __old_kernel_stat *statbuf	Get symbolic link status
108	fstat	unsigned int fd, struct __old_kernel_stat *statbuf	Get file status
114	wait4	pid_t upid, int *stat_addr, int options, struct rusage* ru	Wait for process to change state
115	swapoff	const char *specialfile	Stop swapping
116	sysinfo	struct sysinfo *info	Get system information
117	ipc	unsigned int call, int first, unsigned long second, unsigned long third, void *ptr, long fifth	System V IPC system calls
118	fsync	unsigned int fd	Synchronize file's in-core state
120	clone	unsigned long clone_flags, unsigned long newsp, int	Create child process

Code	Name	Arguments	Description
		parent_tidptr, void tls, int *child_tidptr	
122	uname	struct new_utsname *name	Get system information
125	mprotect	unsigned long start, size_t len, unsigned long prot	Set protection on memory region
126	sigprocmask	int how, old_sigset_t *nset, old_sigset_t* oset	Examine/change blocked signals
132	getpgid	pid_t pid	Get process group ID
133	fchdir	unsigned int fd	Change working directory
140	_llseek	unsigned int fd, unsigned long offset_high, unsigned long offset_low, loff_t *result, unsigned int whence	Reposition 64-bit file offset
141	getdents	unsigned int fd, struct linux_dirent *dirent, unsigned int count	Get directory entries
142	_newselect	int n, fd_set *inp, fd_set* outp, fd_set *exp, struct timeval* tvp	Synchronous I/O multiplexing
143	flock	unsigned int fd, unsigned int cmd	Apply/remove advisory lock
144	msync	unsigned long start, size_t len, int flags	Synchronize memory with physical storage
145	readv	unsigned long fd, const struct iovec *vec, unsigned long vlen	Read from multiple buffers
146	writev	unsigned long fd, const struct iovec *vec, unsigned long vlen	Write to multiple buffers
147	getsid	pid_t pid	Get session ID
148	fdatasync	unsigned int fd	Synchronize file data

Code	Name	Arguments	Description
150	mlock	unsigned long start, size_t len	Lock memory pages
151	munlock	unsigned long start, size_t len	Unlock memory pages
154	sched_setparam	pid_t pid, struct sched_param *param	Set scheduling parameters
155	sched_getparam	pid_t pid, struct sched_param *param	Get scheduling parameters
156	sched_setscheduler	pid_t pid, int policy, struct sched_param *param	Set scheduling policy/parameters
157	sched_getscheduler	pid_t pid	Get scheduling policy
158	sched_yield	-	Yield processor
162	nanosleep	struct timespec *rqtp, struct timespec* rmtp	High-resolution sleep
168	poll	struct pollfd *ufds, unsigned int nfds, long timeout_msecs	Wait for events on file descriptors
172	prctl	int option, unsigned long arg2, unsigned long arg3, unsigned long arg4, unsigned long arg5	Operations on process
174	rt_sigaction	int sig, const struct sigaction *act, struct sigaction* oact, size_t sigsetsize	Examine/change signal action
175	rt_sigprocmask	int how, sigset_t *nset, sigset_t* oset, size_t sigsetsize	Examine/change blocked signals
176	rt_sigpending	sigset_t *set, size_t sigsetsize	Examine pending signals
177	rt_sigtimedwait	const sigset_t *uthese, siginfo_t* uinfo, const struct timespec *uts, size_t sigsetsize	Wait for queued signals
178	rt_sigqueueinfo	pid_t pid, int sig, siginfo_t *uinfo	Queue signal and data

Code	Name	Arguments	Description
179	rt_sigsuspend	sigset_t *unewset, size_t sigsetsize	Wait for signal
180	pread64	unsigned int fd, char *buf, size_t count, loff_t pos	Read from file at given offset
181	pwrite64	unsigned int fd, const char *buf, size_t count, loff_t pos	Write to file at given offset
182	chown	const char *filename, uid_t user, gid_t group	Change file ownership
183	getcwd	char *buf, unsigned long size	Get current working directory
186	sigaltstack	const struct sigaltstack *ss, struct sigaltstack* oss	Set/get signal stack context
187	sendfile	int out_fd, int in_fd, off_t *offset, size_t count	Transfer data between file descriptors
190	vfork	-	Create child process and block parent
192	mmap2	unsigned long addr, unsigned long len, unsigned long prot, unsigned long flags, unsigned long fd, unsigned long pgoff	Map files/devices into memory
195	stat64	const char *filename, struct stat64* statbuf	Get 64-bit file status
196	lstat64	const char *filename, struct stat64* statbuf	Get 64-bit symbolic link status
197	fstat64	unsigned long fd, struct stat64 *statbuf	Get 64-bit file status
199	getuid32	-	Get 32-bit user ID
200	getgid32	-	Get 32-bit group ID
201	geteuid32	-	Get 32-bit effective user ID
202	getegid32	-	Get 32-bit effective group ID

Code	Name	Arguments	Description
220	getdents64	unsigned int fd, struct linux_dirent64 *dirent, unsigned int count	Get 64-bit directory entries
221	fcntl64	unsigned int fd, unsigned int cmd, unsigned long arg	64-bit file control
224	gettid	-	Get thread identification
238	tkill	pid_t pid, int sig	Send signal to thread
240	futex	u32 *uaddr, int op, u32 val, struct timespec* utime, u32 *uaddr2, u32 val3	Fast userspace mutexes
252	exit_group	int error_code	Exit all threads in process
254	epoll_create	int size	Open epoll file descriptor
255	epoll_ctl	int epfd, int op, int fd, struct epoll_event *event	Control epoll file descriptor
256	epoll_wait	int epfd, struct epoll_event *events, int maxevents, int timeout	Wait for epoll file descriptor events
270	tgkill	pid_t tgid, pid_t pid, int sig	Send signal to thread

Quick Reference Categories

File Operations: 3, 4, 5, 6, 8, 10, 15, 19, 38, 39, 40, 41, 42, 63, 85, 92, 93, 94, 95, 106, 107, 108, 141, 143, 145, 146, 148, 180, 181, 182, 183, 187, 195, 196, 197, 220, 221

Process Control: 1, 2, 7, 11, 20, 23, 24, 46, 47, 49, 50, 64, 65, 66, 114, 120, 122, 132, 133, 147, 190, 224, 252

Memory Management: 45, 90, 91, 125, 144, 150, 151, 192

Signal Handling: 37, 48, 67, 72, 73, 126, 172, 174, 175, 176, 177, 178, 179, 186, 238, 270

I/O Multiplexing: 82, 142, 168, 254, 255, 256

Time Operations: 13, 27, 29, 30, 43, 78, 79, 104, 105, 162

System Information: 75, 76, 77, 80, 81, 99, 100, 103, 116