

**NAME**

**sysdecode\_enum**, **sysdecode\_acltype**, **sysdecode\_atfd**, **sysdecode\_extattrnamespace**, **sysdecode\_fadvice**, **sysdecode\_fcntl\_cmd**, **sysdecode\_getfsstat\_mode**, **sysdecode\_getrusage\_who**, **sysdecode\_idtype**, **sysdecode\_ipproto**, **sysdecode\_itimer**, **sysdecode\_kldsym\_cmd**, **sysdecode\_kldunload\_flags**, **sysdecode\_lio\_listio\_mode**, **sysdecode\_madvise**, **sysdecode\_minherit\_flags**, **sysdecode\_msgctl\_cmd**, **sysdecode\_nfssvc\_flags**, **sysdecode\_pathconf\_name**, **sysdecode\_prio\_which**, **sysdecode\_procctl\_cmd**, **sysdecode\_ptrace\_request**, **sysdecode\_rlimit**, **sysdecode\_rtprio\_function**, **sysdecode\_scheduler\_policy**, **sysdecode\_sctp\_pr\_policy**, **sysdecode\_sctp\_sinfo\_flags**, **sysdecode\_semctl\_cmd**, **sysdecode\_shmctl\_cmd**, **sysdecode\_shutdown\_how**, **sysdecode\_sigbus\_code**, **sysdecode\_sigchld\_code**, **sysdecode\_sigfpe\_code**, **sysdecode\_sigill\_code**, **sysdecode\_signal**, **sysdecode\_sigprocmask\_how**, **sysdecode\_sigsegv\_code**, **sysdecode\_sigtrap\_code**, **sysdecode\_sockaddr\_family**, **sysdecode\_socketdomain**, **sysdecode\_sockettype**, **sysdecode\_socketopt\_level**, **sysdecode\_sysarch\_number**, **sysdecode\_umtx\_op**, **sysdecode\_vmresult**, **sysdecode\_whence** - lookup name of various enumerated values

**LIBRARY**

System Argument Decoding Library (libsysdecode, -lsysdecode)

**SYNOPSIS**

```
#include <sysdecode.h>
```

```
const char *
```

```
sysdecode_acltype(int type);
```

```
const char *
```

```
sysdecode_atfd(int fd);
```

```
const char *
```

```
sysdecode_extattrnamespace(int namespace);
```

```
const char *
```

```
sysdecode_fadvice(int advice);
```

```
const char *
```

```
sysdecode_fcntl_cmd(int cmd);
```

```
const char *
```

```
sysdecode_getfsstat_mode(int mode);
```

```
const char *
```

**sysdecode\_getrusage\_who**(*int who*);

*const char \**

**sysdecode\_idtype**(*int idtype*);

*const char \**

**sysdecode\_ipproto**(*int protocol*);

*const char \**

**sysdecode\_itymer**(*int which*);

*const char \**

**sysdecode\_kldsym\_cmd**(*int cmd*);

*const char \**

**sysdecode\_kldunload\_flags**(*int flags*);

*const char \**

**sysdecode\_lio\_listio\_mode**(*int mode*);

*const char \**

**sysdecode\_madvise**(*int advice*);

*const char \**

**sysdecode\_minherit\_flags**(*int inherit*);

*const char \**

**sysdecode\_msgctl\_cmd**(*int cmd*);

*const char \**

**sysdecode\_nfssvc\_flags**(*int flags*);

*const char \**

**sysdecode\_pathconf\_name**(*int name*);

*const char \**

**sysdecode\_prio\_which**(*int which*);

*const char \**

**sysdecode\_procctl\_cmd**(*int cmd*);

*const char \**  
**sysdecode\_ptrace\_request**(*int request*);

*const char \**  
**sysdecode\_rlimit**(*int resource*);

*const char \**  
**sysdecode\_rtprio\_function**(*int function*);

*const char \**  
**sysdecode\_scheduler\_policy**(*int policy*);

*const char \**  
**sysdecode\_sctp\_pr\_policy**(*int policy*);

*const char \**  
**sysdecode\_semctl\_cmd**(*int cmd*);

*const char \**  
**sysdecode\_shmctl\_cmd**(*int cmd*);

*const char \**  
**sysdecode\_shutdown\_how**(*int how*);

*const char \**  
**sysdecode\_sibus\_code**(*int si\_code*);

*const char \**  
**sysdecode\_sigchld\_code**(*int si\_code*);

*const char \**  
**sysdecode\_sigfpe\_code**(*int si\_code*);

*const char \**  
**sysdecode\_sigill\_code**(*int si\_code*);

*const char \**  
**sysdecode\_signal**(*int sig*);

*const char \**

**sysdecode\_sigprocmask\_how**(*int how*);

*const char \**

**sysdecode\_sigsegv\_code**(*int si\_code*);

*const char \**

**sysdecode\_sigtrap\_code**(*int si\_code*);

*const char \**

**sysdecode\_sockaddr\_family**(*int sa\_family*);

*const char \**

**sysdecode\_socketdomain**(*int domain*);

*const char \**

**sysdecode\_sockettype**(*int type*);

*const char \**

**sysdecode\_sockopt\_level**(*int level*);

*const char \**

**sysdecode\_sysarch\_number**(*int number*);

*const char \**

**sysdecode\_umtx\_op**(*int op*);

*const char \**

**sysdecode\_vmresult**(*int result*);

*const char \**

**sysdecode\_whence**(*int whence*);

## DESCRIPTION

The **sysdecode\_enum** functions return a text description of an integer value. The text description matches the name of a C macro with the same value as the sole function argument. NULL is returned if there is no matching C macro name.

Most of these functions decode an argument passed to a system call:

Function	System Call	Argument
----------	-------------	----------

<b>sysdecode_acltype()</b>	acl_get_file(3)	<i>type</i>
<b>sysdecode_atfd()</b>	openat(2)	<i>fd</i>
<b>sysdecode_extattrnamespace()</b>	extattr_get_fd(2)	<i>attrnamespace</i>
<b>sysdecode_fadvice()</b>	posix_fadvise(2)	<i>advice</i>
<b>sysdecode_fcntl_cmd()</b>	fcntl(2)	<i>cmd</i>
<b>sysdecode_getfsstat_mode()</b>	getfsstat(2)	<i>mode</i>
<b>sysdecode_idtype()</b>	procctl(2), waitid(2)	<i>idtype</i>
<b>sysdecode_itimer()</b>	getitimer(2), setitimer(2)	<i>which</i>
<b>sysdecode_kldsym_cmd()</b>	kldsym(2)	<i>cmd</i>
<b>sysdecode_kldunload_flags()</b>	kldunloadf(2)	<i>flags</i>
<b>sysdecode_lio_listio_mode()</b>	lio_listio(2)	<i>mode</i>
<b>sysdecode_madvise()</b>	madvise(2)	<i>advice</i>
<b>sysdecode_minherit_inherit()</b>	minherit(2)	<i>inherit</i>
<b>sysdecode_msgctl_cmd()</b>	msgctl(2)	<i>cmd</i>
<b>sysdecode_nfssvc_flags()</b>	nfssvc(2)	<i>flags</i>
<b>sysdecode_pathconf_name()</b>	pathconf(2)	<i>name</i>
<b>sysdecode_prio_which()</b>	getpriority(2)	<i>which</i>
<b>sysdecode_procctl_cmd()</b>	procctl(2)	<i>cmd</i>
<b>sysdecode_ptrace_request()</b>	ptrace(2)	<i>request</i>
<b>sysdecode_rlimit()</b>	getrlimit(2)	<i>resource</i>
<b>sysdecode_rtprio_function()</b>	rtprio(2)	<i>function</i>
<b>sysdecode_getrusage_who()</b>	getrusage(2)	<i>who</i>
<b>sysdecode_scheduler_policy()</b>	sched_setscheduler(2)	<i>policy</i>
<b>sysdecode_semctl_cmd()</b>	semctl(2)	<i>cmd</i>
<b>sysdecode_shmctl_cmd()</b>	shmctl(2)	<i>cmd</i>
<b>sysdecode_shutdown_how()</b>	shutdown(2)	<i>how</i>
<b>sysdecode_sigprocmask_how()</b>	sigprocmask(2)	<i>how</i>
<b>sysdecode_sockopt_level()</b>	getsockopt(2)	<i>level</i>
<b>sysdecode_sysarch_number()</b>	sysarch(2)	<i>number</i>
<b>sysdecode_umtx_op()</b>	_umtx_op(2)	<i>op</i>
<b>sysdecode_whence()</b>	lseek(2)	<i>whence</i>

These functions decode signal-specific signal codes stored in the *si\_code* field of the *siginfo\_t* object associated with an instance of signal:

<b>Function</b>	<b>Signal</b>
<b>sysdecode_sigbus_code()</b>	SIGBUS
<b>sysdecode_sigchld_code()</b>	SIGCHLD
<b>sysdecode_sigfpe_code()</b>	SIGFPE
<b>sysdecode_sigill_code()</b>	SIGILL

**sysdecode\_sigsegv\_code()** SIGSEGV  
**sysdecode\_sigtrap\_code()** SIGTRAP

Other functions decode the values described below:

**sysdecode\_ipproto()** An IP protocol.

**sysdecode\_sctp\_pr\_policy()** A PR-SCTP policy.

**sysdecode\_signal()** A process signal.

**sysdecode\_sockaddr\_family()** A socket address family.

**sysdecode\_socketdomain()** A socket domain.

**sysdecode\_vmresult()** The return value of a function in the virtual memory subsystem of the kernel indicating the status of the associated request.

## RETURN VALUES

The **sysdecode\_enum** functions return the name of a matching C macro or NULL if no matching C macro was found.

## SEE ALSO

sysdecode(3), sysdecode\_mask(3), sysdecode\_sigcode(3)