NAME

getauxval – retrieve a value from the auxiliary vector

SYNOPSIS

#include <sys/auxv.h>

unsigned long getauxval(unsigned long *type***);**

DESCRIPTION

The **getauxval**() function retrieves values from the auxiliary vector, a mechanism that the kernel's ELF binary loader uses to pass certain information to user space when a program is executed.

Each entry in the auxiliary vector consists of a pair of values: a type that identifies what this entry represents, and a value for that type. Given the argument *type*, **getauxval**() returns the corresponding value.

The value returned for each *type* is given in the following list. Not all *type* values are present on all architectures.

AT_BASE
The base address of the program interpreter (usually, the dynamic linker).

AT_BASE_PLATFORM
A string identifying the real platform; may differ from **AT_PLATFORM** (PowerPC only).

AT_CLKTCK
The frequency with which **times**(2) counts. This value can also be obtained via *sysconf(_SC_CLK_TCK)*.

AT_DCACHEBSIZE
The data cache block size.

AT_EGID
The effective group ID of the thread.

AT_ENTRY
The entry address of the executable.

AT_EUID
The effective user ID of the thread.

AT_EXECFD
File descriptor of program.

AT_EXECFN
Pathname used to execute program.

AT_FLAGS
Flags (unused).

AT_FPUCW
Used FPU control word (SuperH architecture only). This gives some information about the FPU initialization performed by the kernel.

AT_GID
The real group ID of the thread.

AT_HWCAP
An architecture and ABI dependent bit-mask whose settings indicate detailed processor capabilities. The contents of the bit mask are hardware dependent (for example, see the kernel source file *arch/x86/include/asm/cpufeature.h* for details relating to the Intel x86 architecture; the value returned is the first 32-bit word of the array described there). A human-readable version of the same information is available via */proc/cpuinfo*.

AT_HWCAP2 (since glibc 2.18)
Further machine-dependent hints about processor capabilities.

AT_ICACHEBSIZE
The instruction cache block size.

AT_PAGESZ
The system page size (the same value returned by *sysconf(_SC_PAGESIZE)*).

AT_PHDR
The address of the program headers of the executable.

AT_PHENT
The size of program header entry.

AT_PHNUM
The number of program headers.

AT_PLATFORM
A pointer to a string that identifies the hardware platform that the program is running on. The dynamic linker uses this in the interpretation of *rpath* values.

AT_RANDOM
The address of sixteen bytes containing a random value.

AT_SECURE
Has a nonzero value if this executable should be treated securely. Most commonly, a nonzero value indicates that the process is executing a set-user-ID or set-group-ID binary (so that its real and effective UIDs or GIDs differ from one another), or that it gained capabilities by executing a binary file that has capabilities (see **capabilities**(7)). Alternatively, a nonzero value may be triggered by a Linux Security Module. When this value is nonzero, the dynamic linker disables the use of certain environment variables (see **ld-linux.so**(8)) and glibc changes other aspects of its behavior. (See also **secure_getenv**(3).)

AT_SYSINFO
The entry point to the system call function in the vDSO. Not present/needed on all architectures (e.g., absent on x86-64).

AT_SYSINFO_EHDR
The address of a page containing the virtual Dynamic Shared Object (vDSO) that the kernel creates in order to provide fast implementations of certain system calls.

AT_UCACHEBSIZE
The unified cache block size.

AT_UID
The real user ID of the thread.

RETURN VALUE

On success, **getauxval**() returns the value corresponding to *type*. If *type* is not found, 0 is returned.

ERRORS

ENOENT (since glibc 2.19)
No entry corresponding to *type* could be found in the auxiliary vector.

VERSIONS

The **getauxval**() function was added to glibc in version 2.16.

ATTRIBUTES

For an explanation of the terms used in this section, see **attributes**(7).

Interface	Attribute	Value
getauxval()	Thread safety	MT-Safe

CONFORMING TO

This function is a nonstandard glibc extension.

NOTES

The primary consumer of the information in the auxiliary vector is the dynamic linker, **ld-linux.so**(8). The auxiliary vector is a convenient and efficient shortcut that allows the kernel to communicate a certain set of standard information that the dynamic linker usually or always needs. In some cases, the same information could be obtained by system calls, but using the auxiliary vector is cheaper.

The auxiliary vector resides just above the argument list and environment in the process address space. The auxiliary vector supplied to a program can be viewed by setting the **LD_SHOW_AUXV** environment variable when running a program:

```
$ LD_SHOW_AUXV=1 sleep 1
```

The auxiliary vector of any process can (subject to file permissions) be obtained via */proc/[pid]/auxv*; see **proc**(5) for more information.

BUGS

Before the addition of the **ENOENT** error in glibc 2.19, there was no way to unambiguously distinguish the case where *type* could not be found from the case where the value corresponding to *type* was zero.

SEE ALSO

secure_getenv(3), **vdso**(7), **ld-linux.so**(8)

COLOPHON

This page is part of release 5.02 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at https://www.kernel.org/doc/man−pages/.