NAME

w – who is logged in, and what are they doing

SYNOPSIS

w [**–hlsuw**] [*user*]

AVAILABILITY

SUNWcsu

DESCRIPTION

The **w** command displays a summary of the current activity on the system, including what each user is doing. The heading line shows the current time, the length of time the system has been up, the number of users logged into the system and the average number of jobs in the run queue over the last 1, 5 and 15 minutes.

The fields displayed are: the users login name, the name of the tty the user is on, the time of day the user logged on (in *hours:minutes*), the idle time—that is, the number of minutes since the user last typed anything (in *hours:minutes*), the CPU time used by all processes and their children on that terminal (in *minutes:seconds*), the CPU time used by the currently active processes (in *minutes:seconds*), the name and arguments of the current process.

If a *user* name is included, output is restricted to that user.

OPTIONS

–h Suppress the heading.

–l Produce a long form of output, which is the default.

–s Produce a short form of output. In the short form, the tty is abbreviated, the login time and CPU times are left off, as are the arguments to commands.

–u Produces the heading line which shows the current time, the length of time the system has been up, the number of users logged into the system, and the average number of jobs in the run queue over the last 1, 5 and 15 minutes.

–w Produces a long form of output, which is also the same as the default.

EXAMPLE

```
example% w
10:54am  up 27 day(s), 57 mins,  1 user,  load average: 0.28, 0.26, 0.22
User     tty           login@    idle   JCPU   PCPU   what
ralph    console       7:10am       1   10:05   4:31   w
```

ENVIRONMENT

If any of the **LC_*** variables (**LC_CTYPE, LC_MESSAGES, LC_TIME, LC_COLLATE, LC_NUMERIC,** and **LC_MONETARY**) (see **environ**(5)) are not set in the environment, the operational behavior of **tar** for each corresponding locale category is determined by the value of the **LANG** environment variable. If **LC_ALL** is set, its contents are used to override both the **LANG** and the other **LC_*** variables. If none of the above variables is set in the environment, the "C" (U.S. style) locale determines how **tar** behaves.

LC_CTYPE

Determines how **tar** handles characters. When **LC_CTYPE** is set to a valid value, **tar** can display and handle text and filenames containing valid characters for that locale. **tar** can display and handle Extended Unix code (EUC) characters where any individual character can be 1, 2, or 3 bytes wide. **tar** can also handle EUC characters of 1, 2, or more column widths. In the "C" locale, only characters from ISO 8859-1 are valid.

LC_MESSAGES
Determines how diagnostic and informative messages are presented. This includes the language and style of the messages, and the correct form of affirmative and negative responses. In the "C" locale, the messages are presented in the default form found in the program itself (in most cases, U.S. English).

LC_TIME
Determines how **tar** handles date and time formats. In the "C" locale, date and time handling follow the U.S. rules.

FILES

/var/adm/utmp

SEE ALSO

ps(1), **who**(1), **whodo**(1M), **utmp**(4)

NOTES

The notion of the ‘‘current process’’ is muddy. The current algorithm is ‘the highest numbered process on the terminal that is not ignoring interrupts, or, if there is none, the highest numbered process on the terminal’. This fails, for example, in critical sections of programs like the shell and editor, or when faulty programs running in the background fork and fail to ignore interrupts. In cases where no process can be found, **w** prints –.

The CPU time is only an estimate, in particular, if someone leaves a background process running after logging out, the person currently on that terminal is ‘‘charged’’ with the time.

Background processes are not shown, even though they account for much of the load on the system.

Sometimes processes, typically those in the background, are printed with null or garbaged arguments. In these cases, the name of the command is printed in parentheses.

w does not know about the conventions for detecting background jobs. It will sometimes find a background job instead of the right one.