

NAME

remark, vremark – display formatted informational notification

SYNOPSIS

```
#include "mjsu.h"
```

```
VOID remark(CHAR *fmt, ...);
VOID vremark(CHAR *fmt, va_list ap);
```

DESCRIPTION

If *fmt* is not NULL, **remark()** and **vremark()** construct an informational notification message and then display it.

The *message* is constructed from the format-specification *fmt* using subsequent arguments, in the same way as the standard **fprintf(3)** and **vfprintf(3)** functions, respectively.

By default, the notification is emitted to standard-error, prefixed by the name of the current process (see **whatami(3)**), and followed by a newline.

That default method of display can be overridden using **notifier(3)** to install a callback function that displays the notification in another manner (eg: as a popup dialog with a custom icon, or whatever).

RETURNS

remark() and **vremark()** return the number of times that either function has been called with a non-NULL *fmt*, including the current call, ie: the total number of such informational notifications issued so far.

EXAMPLE

```
INT n;

for (n = 1; n <= 5; ++n)
{
    remark("job %i starting", n);
    initialise_job(n);

    remark("job %i continuing", n);
    run_job(n);
}
remark("processing complete");
```

SEE ALSO

warning(3), **error(3)**, **usage(3)**.

notifier(3), **whatami(3)**, **mjsu(7)**.

exit(), **fprintf()**, **vfprintf()**, as defined by ANSI X3.159-1989.

AVAILABILITY

These functions are written in C, conforming to ANSI X3.159-1989 (hosted program environment).