

**NOM**

write – Écrire dans un descripteur de fichier.

**SYNOPSIS**

```
#include <unistd.h>
```

```
ssize_t write(int fd, const void *buf, size_t count);
```

**DESCRIPTION**

**write()** écrit jusqu'à *count* octets dans le fichier associé au descripteur *fd* depuis le tampon pointé par *buf*. POSIX réclame qu'une lecture avec **read()** effectuée après le retour d'une écriture avec **write()**, renvoie les nouvelles données. Notez que tous les systèmes de fichiers ne sont pas compatibles avec POSIX.

**VALEUR RENVOYÉE**

**write()** renvoie le nombre d'octets écrits (0 signifiant aucune écriture), ou  $-1$  s'il échoue, auquel cas *errno* contient le code d'erreur. Si *count* vaut zéro, et si le descripteur est associé à un fichier normal, 0 sera renvoyé sans effets de bord. Pour un fichier spécial, les résultats ne sont pas portables.

**ERREURS****EAGAIN**

L'écriture est non-bloquante (attribut **O\_NONBLOCK** du descripteur), et l'opération devrait bloquer.

**EBADF**

*fd* n'est pas un descripteur de fichier valide, ou n'est pas ouvert en écriture.

**EFAULT**

*buf* pointe en dehors de l'espace d'adressage accessible.

**EFBIG**

Tentative d'écrire un fichier dont la taille dépasse un maximum dépendant de l'implémentation ou du processus, ou d'écrire à une position qui dépasse le déplacement maximal autorisé.

**EINTR**

L'appel système a été interrompu par un signal avant d'avoir pu écrire quoique ce soit.

**EINVAL**

*fd* correspond à un objet ne permettant pas l'écriture ; ou le fichier a été ouvert avec l'attribut **O\_DIRECT**, et soit l'adresse spécifiée dans *buf*, soit la valeur spécifiée dans *count*, soit la tête de lecture du fichier ne sont pas correctement alignés.

**EIO** Une erreur d'entrée-sortie bas niveau s'est produite durant la modification de l'i-noeud.

**ENOSPC**

Le périphérique correspondant à *fd* n'a plus de place disponible.

**EPIPE** *fd* est connecté à un tube (pipe) ou une socket dont l'autre extrémité est fermée. Quand ceci se produit, le processus écrivain reçoit un signal **SIGPIPE**. S'il intercepte, bloque ou ignore ce signal, **EPIPE** est renvoyé.

D'autres erreurs peuvent se produire suivant le type d'objet associé à *fd*.

**CONFORMITÉ**

SVr4, BSD 4.3, POSIX.1-2001.

Sous SVr4, un appel **write()** peut être interrompu, et renvoyer **EINTR** à n'importe quel moment, pas seulement avant l'écriture des données.

**NOTES**

Une réussite de **write()** n'offre aucune garantie que les données se trouvent sur le disque. En fait, sur certaines implémentations erronées, elle ne garantit même pas que l'espace suffisant a été réservé pour les données. La seule manière d'être sûr de la réussite est d'invoquer **fsync(2)** après avoir écrit les données.

**VOIR AUSSI**

**close(2), fcntl(2), fsync(2), ioctl(2), lseek(2), open(2), pwrite(2), read(2), select(2), writev(2), fwrite(3)**

**TRADUCTION**

Ce document est une traduction réalisée par Christophe Blaess <<http://www.blaess.fr/christophe/>> le 12 octobre 1996 et révisée le 14 août 2006.

L'équipe de traduction a fait le maximum pour réaliser une adaptation française de qualité. La version anglaise la plus à jour de ce document est toujours consultable via la commande : « **LANG=C man 2 write** ». N'hésitez pas à signaler à l'auteur ou au traducteur, selon le cas, toute erreur dans cette page de manuel.