

**NOM**

`read` – Lire le contenu d'un fichier.

**SYNOPSIS**

```
#include <sys/types.h>
#include <unistd.h>
```

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

**DESCRIPTION**

`read()` lit jusqu'à *count* octets depuis le descripteur de fichier *fd* dans le tampon pointé par *buf*.

Si *count* vaut zéro, `read()` renvoie zéro et n'a pas d'autres effets. Si *count* est supérieur à `SSIZE_MAX`, le résultat est indéfini.

**VALEUR RENVOYÉE**

`read()` renvoie `-1` s'il échoue, auquel cas *errno* contient le code d'erreur, et la position de la tête de lecture est indéfinie. Sinon, `read()` renvoie le nombre d'octets lus (0 en fin de fichier), et avance la tête de lecture de ce nombre. Le fait que le nombre renvoyé soit plus petit que le nombre demandé n'est pas une erreur. Ceci se produit à la fin du fichier, ou si on lit depuis un tube ou un terminal, ou encore si `read()` a été interrompu par un signal.

**ERREURS****EAGAIN**

On utilise une lecture non bloquante (attribut `O_NONBLOCK` du descripteur de fichier) et aucune donnée n'était disponible.

**EBADF**

*fd* n'est pas un descripteur valide, ou n'est pas ouvert en lecture.

**EFAULT**

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

D'autres erreurs peuvent se produire, suivant le type d'objet associé à *fd*. POSIX permet à un `read()` interrompu par un signal de renvoyer soit le nombre d'octets lus à ce point, soit `-1`, et de placer *errno* à `EINTR`.

**EINTR**

`read()` a été interrompu par un signal avant d'avoir eu le temps de lire quoi que ce soit.

**EINVAL**

*fd* correspond à un objet ne permettant pas la lecture ; 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**

Erreur d'entrée-sortie. Ceci arrive si un processus est dans un groupe en arrière-plan et tente de lire depuis le terminal. Il reçoit un signal `SIGTTIN` mais il l'ignore ou le bloque. Ceci se produit également si une erreur d'entrée-sortie bas-niveau s'est produite pendant la lecture d'un disque ou d'une bande.

**EISDIR**

*fd* est un répertoire.

**CONFORMITÉ**

SVr4, BSD 4.3, POSIX.1-2001.

**RESTRICTIONS**

Sur un système de fichiers NFS, la lecture de petites quantités de données ne mettra à jour l'horodatage du fichier que lors de la première lecture. Les lectures suivantes ne modifieront pas cette heure. En effet la plupart, si ce n'est tous les clients NFS disposent d'un cache des attributs de fichiers et n'effectuent pas la mise à jour du champ « `st_atime` » (horodatage du dernier accès au fichier) du côté serveur. La véritable sémantique UNIX peut être obtenue en désactivant le cache des attributs du côté client, mais généralement ceci augmente sensiblement la charge du serveur, et dégrade ses performances.

Beaucoup de disques et systèmes de fichiers ont été considérés comme suffisamment rapides pour que l'implémentation de **O\_NONBLOCK** soit jugée inutile. Ainsi cet attribut peut ne pas être disponibles sur les fichiers et/ou les disques.

**VOIR AUSSI**

**close(2), fcntl(2), ioctl(2), lseek(2), open(2), pread(2), readdir(2), readlink(2), readv(2), select(2), write(2), fread(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 read** ». N'hésitez pas à signaler à l'auteur ou au traducteur, selon le cas, toute erreur dans cette page de manuel.