

**Programmation
système en C
sous Linux**
Signaux, processus,
threads, IPC et sockets
2^e édition

Programmation système en C sous Linux: Signaux, processus, threads, IPC et sockets

- Author : Christophe Blaess
- Publisher : Eyrolles, 2011
- pages : 964 pages
- N° Class : 621/574



Tirer le meilleur parti de l'environnement Linux

La possibilité de consulter les sources du système, de la bibliothèque glibc et de la plupart des applications qui tournent sur cet environnement représente une richesse inestimable aussi bien pour les passionnés qui désirent intervenir sur le noyau, que pour les développeurs curieux de comprendre comment fonctionnent les programmes qu'ils utilisent quotidiennement.

Nombreuses sont les entreprises qui ont compris aujourd'hui tout le parti qu'elles pouvaient tirer de cette ouverture des sources, gage de fiabilité et de pérennité, sans parler de l'extraordinaire niveau de compétences disponible au sein d'une communauté de programmeurs aguerris au contact du code des meilleurs développeurs OpenSource.

Un ouvrage conçu pour les programmeurs Linux et Unix les plus exigeants

Sans équivalent en langue française, l'ouvrage de Christophe Blaess constitue une référence complète de la programmation système sous Linux, y compris dans les aspects les plus avancés de la gestion des processus, des threads ou de la mémoire. Les programmeurs travaillant sous d'autres environnements Unix apprécieront tout particulièrement l'attachement de l'auteur au respect des standards (C Ansi, glibc, Posix...), garant d'une bonne portabilité des applications. La deuxième édition de ce livre a été entièrement actualisée en fonction du noyau Linux 2.6 et des nombreuses évolutions de la bibliothèque C.

À qui s'adresse cet ouvrage ?

- Aux programmeurs et développeurs intéressés par les aspects système de la programmation sous Linux et Unix.
- Aux administrateurs système en charge de la gestion d'un parc Linux et/ou Unix.
- Aux étudiants en informatique (1e et 2e cycle universitaire, écoles d'ingénieurs, etc.).