

1 Droits d'accès et masque binaire

Dans les systèmes d'exploitation dérivé d'Unix, le codage des droits se fait sur 9 bits groupés par 3 bits qui sont dans l'ordre : r le fichier est lisible par le propriétaire (- dans le cas contraire); w le fichier est modifiable par le propriétaire (- sinon); x le fichier est exécutable par le propriétaire (- sinon). Le groupe suivant de 3 bits reprend le même principe mais définit les droits pour les membres du groupe auquel appartient le propriétaire et le dernier groupe concerne les autres utilisateurs suivant le même schéma. Ces droits sont codés en un entier. Pour ce faire, on convient de la correspondance : $r = 4$, $w = 2$ et $x = 1$. Ainsi, les droits rw-, correspondent à l'entier $(r =)4 + (w =)2 = 6$. Donc $rw - rw - rw -$ correspond à l'entier 666.

Questions

1. A quels droits correspondent les entiers 751, 521, 214 et 150 ?
2. Par quels entiers sont codés les droits rw-r- -r- - et rwxr-xr-x ?

Droits par défaut

Par ailleurs, à la création d'un fichier, des droits d'accès par défaut sont donnés à ce fichier. En standard, l'entier codant ces droits est 666 pour un fichier et 777 pour un répertoire. La commande *umask* permet de définir la protection maximale que l'on donne au cours de la session aux nouveaux fichiers; sa syntaxe est : *umask joctalj*. Les droits d'accès sont déterminés après l'opération logique suivante : défaut ET (NON octal). Plus simplement et lorsque c'est possible, on part des droits par défaut et on retranche le masque octal. Si ce n'est pas possible, on applique la règle précédente. Pour connaître la valeur du masque courant, il suffit d'utiliser la commande *umask* sans arguments.

Questions

1. A quels droits correspondent les masquages 77 et 244 dans le cas d'un fichier et dans celui d'un répertoire ?
2. . Quels masques faudrait-il appliquer afin d'obtenir pendant la session les droits rw-rw-r- - à chaque création de fichier dans le cas d'un fichier et dans celui d'un répertoire ?

2 Stratégies de stockage des fichiers

Un fichier est divisé en blocs. Nous allons expliciter quelques points concernant le choix de la taille de ces derniers.

1. On montre que la taille moyenne des fichiers UNIX est de 1Ko. En allouant une taille de 32Ko à chaque bloc, quel pourcentage de l'espace disque est utilisé ?

2. Si on représente les blocs libres par une liste chaînée et que l'on suppose que les blocs font 1Ko alors que leurs numéros physiques sont codés sur 16bits, de combien de blocs a besoin un disque de 20Mo pour contenir tous les numéros physiques des blocs libres ?
3. Même question si on représente les blocs libres par une table de bits (une bitmap).
4. Dans ce contexte, si on représente un fichier par une liste chaînée de blocs et que l'on désire lire l'octet 215, combien de blocs doit on parcourir ?
5. Si on représente les blocs libres par une table de bits, quelle serait la taille de cette table pour un disque de 1.3Go avec des blocs de 512Ko.
6. Dans ce codage, donner la formule qui permet d'avoir le premier bloc libre.