cpFS Permission System

--[cppig1995]

The 'cpFS' file system uses its own permission system called "cpFS-PS (an abbreviation for 'cpFS Permission System')". This system uses 16 bits; 2 octets to represent the entry's permission. The 16 attributes are as follows :

b s l d r c x a r c x a r c x a

The bit 'b' means this entry is broken.
A broken file can only opened by a binary with the kernel permission; like a file recovery softwares.
The bit 's' means this entry is sticky; without remove this bit from this entry, you cannot delete this file.
The bit 'l' means this entry is a soft link to another file or directory.
The bit 'd' means this entry is a directory.
And we can see 'r c x a', exactly 3 times.
The first one is for owner. Second one is for group, and third one is for others.
The bit 'r' means that we can read this entry.
The bit 'c' means that we can change or append to this entry.
The bit 'x' means that we can execute this entry.
The bit 'a' means that we can only append to

this entry, without change its original contents.
  So we can represent the permission of an entry like this :

  ----rcxar-x-----

  This entry is not broken, not sticky, not a link, not a directory(so a file), and owner can do all the things (rcxa), and group can read and execute the file (r-x-), and others can't access (----).
  Like this example, we can represent 0 value with hyphen('-'), and 1 value with the name of the bit.

  We can also use like this : *to give the permission is '+?', to take is '-?', and take the all permission and give only specified is '=?', etc.* In example, to take a read permission is '-r', to give a change permission is '+c', to take all and give only execute permission is '=x'.
  And, we can find a small bug. We have 3 'r' bits, 3 'c' bits, 3 'x' bits, 3 'a' bits, and we must access 3 bit with 1 symbol.
  Therefore, we can fix this bug with tag a prefix to a symbol's name as the followings :
  The prefix for owner is 'o'(owner), for group is 'g'(group), for other is 'e'(else). And, to change all 3 bits, we can use 'a'(all). To specify more than 1 parts, we can use all of necessary symbols like 'og' (owner and group), 'ge'(group

and others).

For example, *'r' bit for group is 'gr', 'c' bit for owner is 'oc', 'x' bit for others is 'ex', 'a' bit for owner is 'oa', etc.* The prefix which tagged in 'b', 's', 'l', 'd' bits are ignored.

So we can use like this : *'+gr', '-oc', '=ex'*.

We can also use like this : *'g+r', 'o-c', 'e=x'*.

The second notation is useful when we change the permission of more than 1 parts the same. Example is : *'og+r', 'ge-c', 'ge='*.

As you see, if nothing is right of '+' or '-', it is processed as an error. But if nothing is right of '=' symbol, it processed as take all permission from the specified part.

We can also use a hexadecimal(16-based) representation for permission. We must use 4 numbers to represent the permission.

4xxx means sticky, 0xxx means non-sticky.

xnxx means owner has n permission.

xxnx means group has n permission.

xxxn means others has n permission.

We can calculate n by this method :

'r' is 8, 'c' is 4, 'x' is 2, 'a' is 1 and '-' is 0.

Adding all of the numbers and the hexadecimal reprentation of the result is 'n'.

In example, *'r-xa'* is *8+0+2+1 = 11 = B.*

The total example of this representation method are following :

Non-sticky : so 0.
Owner : 'rcxa' = 8+4+2+1 = 15 = F.
Group : 'r-xa' = 8+0+2+1 = 11 = B.
Others : 'r-x-' = 8+0+2+0 = 10 = A.

So Hexadecimal representation of
  '----rcxar-xar-x-'
is *0FBA.*

And we can use 8, 2, 1 in first number : '*b*' is 8, '*l*' is 2, '*d*' is 1.
But we cannot change broken into non-broken, link into non-link, directory into file, so in first number, we cannot use 8, 2, 1 to a hexadecimal representation which is to change a permission of an entry, but we can use 8, 2, 1 only to a hexadecimal representation which is to display permission status of an entry.

In this permission system, the owner of the entry can change the permission freely – *without broken bit* – even it has no permission for owner to read, change, execute or append.
Additionally, in these contexts, 'entry' means both files and directories (include links; because they are only a tricked file.)