COMP 7500/7506 Advanced Operating Systems

Project 4: cpmFS - A Simple File System

Frequently Asked Questions

Document History: Created on April 25, 2019. Revised on April 25, 2019. **Version 1.0**

1. In cpmfsys.h file there are 5 functions at the bottom:
   - `cpmCopy`
   - `cpmOpen`
   - `cpmClose`
   - `cpmRead`
   - `cpmWrite`

   which says that need not to be implemented for project 4. Are they need to be implemented in project 4?

Answer: You don't need to implement the above five functions in project 4. Your source code file `cpmfsys.c` should include the implementations of the following nine functions:

- `mkDirStruct()`
- `writeDirStruct()`
- `makeFreeList()`
- `printFreeList()`
- `cpmDir()`
- `checkLegalName()`
- `findExtentWithName()`
- `cpmDelete()`
- `cpmRename()`

2. Are we supposed to use `diskSimulator.c` and other files from the course repository or are we supposed to build our own version of all of those files?

Answer: You only need to implement `cpmfsys.c` by making use of the existing `diskSimulator.c` and the other files from the course repository.

3. Do we ignore the actual execution time and use the execution time given by the run command? If the last sector is fully used, would it be save to assume that RC would have the

value of 8? This would indicate that all 1024 (8*128) bytes have been used. However, so far I do not see any RC with the value of 8.

Answer: RC is the number of sectors in the last partially-filled block. The maximum possible RC should be 7.

4. Regular expression aren't part of ANSI C. But I found that regex.h can do regular expression which can help us check the file name more easily. Can we use this header file?

Answer: You are allowed to include `regex.h` in your implementation.

5. When I try to inspect the disk image, it appears the only block with contents is block 0. Is this correct? Since block 0 refers to locations for file contents, I expected to find that actual content elsewhere in the disk image. It seems the project can be completed without having this content, but I was just checking to see if I was missing something.

Answer: You must exam the extents in block 0 to determine if there are allocated and used blocks (i.e., block address of 1-255). The information can be found in the last 16 bytes of each extent. I agree with you that this project might be completed without having the data content (i.e., blocks are allocated without containing any content).

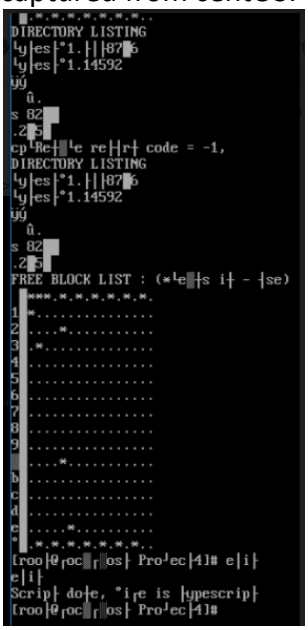
6. Why users must specify a job's CPU time priori to the job submission? Function cpmDir prints weired output. It also caused the centOS system a lot of errored displays. The solutions I tried as follows:
   1) I tried to download the image1.img again, and the error still shows.
   2) If I only print the first 5 lines of image1.img (or the first two files), no error in the output.
   3) If I use script to save the output, centOS error display are correct in typescript file, but the weired output for the file names are still there.

   Does anyone have the same issue?

   In saved typescript:

```
DIRECTORY LISTING
mytestf1.txt 8706
mytestf1. tx 14592
ÿýû.
s 8200
01 . 2050
```

Direct captured from centOS:



Answer: Your `cpmDir()` function might have a bug. Prior to printing an extent (say `d`) in block 0, your `cpmDir` must check the `status` of the extend (i.e., `d->status != 0xe5`). If `d->status == 0xe5`, then this extent is invalid. Don't print the extent if it is invalid.