

Operating Systems HW #2

Due 31.3.2015, 23:55

In this assignment you will implement a **RAID-5** simulator.

Based upon the **RAID-0** simulator seen in class, you will extend the available operations and match the properties of RAID-5.

To implement the simulator, you will add/modify the following commands and make the following changes:

- Add a **REPAIR** command.
The REPAIR command will receive a disk index as argument, similar to the KILL command. The given command will simulate repairing or replacing a device, by simply marking it as no longer faulty.
If the previous device is *not* faulty, the operation should still succeed as usual and the device replaced / reopened.
- Replace the **OP** command with **READ** and **WRITE** commands.
The command will translate a single logical sector (the command parameter) to a physical sector and device by printing an output matching the output in the example simulator exactly.
Note that some operations might require access to several sectors on several devices – in such cases, create a new output line for each sector accessed in each device, ordered according to the device index, then sector index.
If one or more sectors are in faulty devices and **cannot be restored**, print the output regularly until a faulty device is encountered; once encountered, print the same error output as in the example simulator, and **cancel** the operation, i.e., do not continue translating and outputting further sectors for this operation.
To reiterate, an operation should not fail if the data can be read/written by utilizing other devices!
- Add actual devices to the simulator.
The program will receive device names as arguments, **instead** of the number of devices. The devices should be opened when the program starts, and closed properly once it finishes (e.g., `./raid5 /dev/sdb /dev/sdc /dev/sdd`).
In addition to printing the output of the translation, each read/write operation should be preceded by an actual read/write from the relevant device. On a REPAIR command, open (or reopen) the device provided originally by the program's command-line arguments.
Assume a sector size of 4KB. You should ignore the actual data read/written (no need to XOR or interpret it in any way).
- Make sure your code works with any block size!
- Finally, we will address errors in the simulator.
In any error accessing a device (read, write, open, etc.) – do not terminate the program. Instead, treat the disk as failed, output a proper error message, and continue operating as if the user issued a KILL command on that disk.
This is true for the REPAIR command as well – if the new device is faulty and thus it fails to open, it will not be replaced, and the RAID-5 structure will be left as it previously was.
- **DO NOT CHANGE ANY OUTPUT FROM THE GIVEN SIMULATOR!**

The program should always exit cleanly when done, i.e. “faulty” RAID disks are not considered errors – but you should still output a proper error message on any fault (except the KILL command / `close()` system call).

Guidelines

- Use CTRL+D to send EOF to the console and exit gracefully.
- Parse and output everything exactly matching the provided code from the recitation.
- Use only system calls to access the devices. You can use standard C functions to read user input only (`stdin`).
- You can assume correctness of input – all devices exist, of same size and have enough capacity, input is valid, etc.
- If you fail opening, reading, writing, or closing a device – treat it as failed (**KILL**), output a proper message, and continue.
- Submit: a single C file **raid5.c**