Team 22
s202897 Vermeylen Clément &
s181482 Baguette Brice

Overview of your approach

First, we design a structure that has the field that was needed to store the information for each processes. It has those fields:

- pid[10]: Used to store the pids for a set of processes that have the same name. At first we implemented it with dynamical allocation but it leads to crash our VM with a NULL pointer
- total_pages: use to store the amount of total pages.
- valid_pages: use to store the amount of valid pages. To retrieve those, we will iterate over the memory info to look if the pte_present flag of the page is there. If that's the case, the page is therefore on the RAM.
- invalid_pages: use to store the amount of unvalid pages.
- nb_group: use to store the number of groups
- identical_page_groups: use to store the number of different pid that has the have the same page group
- may_be_shared: amount of pages that may be shared
- name[255]: the name of the process

When the module is initiated, we populate a list of this structure with the information of all processes. Checking if a process with the name is already in our list and depending on that we modify the relative information for the set of processes or we append a new item to our list. Therefore we will also use the detection of identical pages. To do this, we iterate over the PIDs of the same process name, retrieving each page. To check whether one page is identical to another, we use a hash table. If a page is not in the table, we hash the data and add the hash to the table. Otherwise, if it is already in the table, we increment the *may_be_shared* variable and check the flag. If the flag is false, this means that the first two pages are identical. We must therefore also increment *may_be_shared* and *nb_group*. If the flag is true, we do nothing more

Then when we will write in the /proc/memory_info file using echo command line, we will parse the file to detect what's the command that has been put in. Depending there we will either write every info with the good format in the file or only one information. We will also remove a process info from the data structure if we use DEL command and reset our data structure and therefore re initialize it when we use the RESET command.

Feedback

- Difficulty: It was easier to start than the previous project since we had the tutorial 4 as starting point to understand how we should work it out. It was still a bit tricky and we spend a lot of time to detect the valid pages in the good way or for the detection of identical pages.
- Amount of work: Around 30 hours per person
- Other: Compare to the other project, it was easier to progress step by step and see the result of the code, which is a good thing to understand and progress.