VE482 — Introduction to Operating Systems

Lab 8
Manuel — UM-JI (Fall 2020)

Goals of the lab

- Understand the basics about memory

- Compare two page replacement algorithms

- Work with Minix 3 kernel source code

1 Introduction

You have handed in your `mumsh` a long time ago, but your mum is still complaining about your work: "You were too slow! and you almost missed some milestones", she said, "If you keep being this slow, you will get fired by Lemonion Inc.." You immediately try to explain that Mr. Frown is making your life a nightmare, but she cuts you short: "Stop making up stories to excuse your slowness and bad work quality! I met Mr. Frown over a week ago when I came to bring your lunch box that you had forgotten and he is a real gentlemen. He was very kind, he even offered me a coffee and a piece of cake. We had a friendly conversation and to be honest he was not surprised to hear that you were spending all your time playing video games... He comforted me, saying that he subdued worse kids than you and you will soon be back on track, he would personally take care of you! I cannot believe a single word of all the mean things you say about that selfless man. You should be ashamed of slandering such a nice person, and instead should be thankful for everything he does for you. This is not how I have raised you!"

Your mum is obviously very upset at you, and on those last words she leaves you alone with your thoughts. This double face Mr. Frown managed to even ruin your family time. How can someone be so evil? You have no idea how much longer you will be able to cope with this situation. It was still acceptable when you were in peace at home, but now with mum on your back that really feels terrible. In any case you now understand why things are getting worse and worse with Mr. Frown; that was not a mere feeling, he really is meaner than ever with you!

As you ponder over what has just happened you find so many arguments why mum is wrong about you being too slow with `mumsh`. First you had very little knowledge about shell and terminal when she assigned you the task, then C is a language you learned many years ago and almost never used since then. In fact this is your Least Recently Used (LRU) language, so it makes sense that you have forgotten most of it.

It feels like what mum would have wanted from you is to apply a strategy where you keep the LRU information and forget about the Most Recently Used (MRU) one. Feeling upset you want to prove to your mum that she is wrong. After a bit of thinking you remember having read somewhere that the most widely used OS worldwide, Minix 3, is using LRU for handling page faults. If you could adjust Minix 3 memory management such that MRU is used instead then you could prove that Mum's Really Unfair! As you feel pretty proud of your pun you move on to investigate Minix 3 memory management.

2 Memory in Minix 3

Although Minix 3 micro kernel has very small size compared to Linux or Windows kernels it requires a bit of work to get use to the kernel coding style. So first you will investigate memory in general and then test your theory.

2.1 Memory management at kernel level

To better grasp the importance of memory in an OS and become familiar with the structure of Minix 3 kernel you ask yourself the following questions.

- What does `vm` stands for? (Hint: in this context the answer is not virtual machine)

- Find all the places where the `vm` used inside the kernel, Why does it appear in so many different places?

- How is memory allocated within the kernel? Why are not `malloc` and `calloc` used?

- While allocating memory, how does the functions in kernel space switch back and fro between user and kernel spaces? How is that boundary crossed? How good or bad it is to put `vm` in userspace?

- How are pagefaults handled?

2.2 Mum's Really Unfair!

Now that you have a better idea of how important memory is in an OS and how it is implemented in Minix 3, it is time for you to make your point: lets prove that MRU is worse than LRU!

- What algorithm is used by default in Minix 3 to handle pagefault? Find its implementation and study it closely.

- Use the `top` command to keep track of your used memory and cache, then run `time grep -r "mum" /usr/src`. Run the command again. What do you notice?

- Adjust the implementation of LRU into MRU and recompile the kernel.

- Use the `top` command to keep track of your used memory and cache, then run `time grep -r "mum" /usr/src`. Run the command again. What do you notice?

- Discuss the different behaviours of LRU and MRU as well as the consequences for the users. Can you think of any situation where MRU would be better than LRU?

You are proud of your achievement, Mum's Really Unfair and Mr. Frown is now your worst enemy. While you know that you should sleep because it is very late, you still want to check an extra thing: How do LRU and MRU compare when many processes allocate much static and dynamic memory, and then recall it? This is an interesting question you might want to answer but can chose to skip in order to enjoy a good night sleep. You definitely need one as tomorrow you are going to face an even worse Mr. Frown. As for now there is no point on focusing on such negative thoughts such you prefer to relish on your victory.