Readme for Assignment 3

We started out by essentially following the assignment specification.

mprotect

Adding the system calls was somewhat easy as we simply did a linear search of the page table using walkpgdir() to find the relevant memory address' page table and mark the corresponding PTE_ bits using bitwise operations.

Most of our knowledge of how to implement the page table search came from the copyuvm() function.

Copy On Write

Copy on write was significantly more challenging, what we needed to do was create hollow pointers to the same page table entry for the child process.

That shared page table was then marked read only, here is the reason, when we mark the page table read only we will get a page fault when either of the processes (the parent of the child) would try to write.

When one of those processes creates the page fault we can intercept this in trap.c and look at the control register to find the specific address that the process was trying to write to.

Knowing the address we can then use the page table's roundown function to find the beginning of that particular page table entry.

Once we are at the address pointing to the beginning of that entry, we know that the first 20 bytes are going to be the PPN, the physical page number. At which point we can create a new page for the process attempting to write and give that page it's own copy of whatever is at that address in physical memory. (a 4096 bit chunk of memory.)

Thank you.

Abanoub David Awad -- ada80 Nick Girardo -- nag82