

Lab 2: Copy on Write Lab

Copy-on-write (COW) is a key operating systems functionality that avoids unnecessary copying of memory when a new process is created (via `fork()`). The parent and child are supposed to have the same memory layout after a `fork()` call, but copying the memory of the parent to the child expensive and often wasteful. Think about why it wasteful. As a result, the COW mechanism was developed. COW copies only the page mappings for the parent process (i.e., from the parent's pagetable) to the child process. Since the parent and child process may diverge after both start executing, the pages are marked as read-only in the pagetable and copied to a new physical frame when a write operation is requested against the shared page.

Please implement COW to handle the parent and child execution from the creation of the child, through all memory operations of the parent and child (including writes), and accounting for the deallocation of physical frames (e.g., when either the child or parent terminates).

What to submit:

You need to submit the following:

- (1) A diff of xv6 code you have modified (hint: use “git diff”)
- (2) A detailed explanation what changes you have made and necessary screenshots to show your work and results
- (3) A detailed description of XV6 source code (including your modifications) about how the info system call is processed, from the user-level program into the kernel code, and then back into the user-level program.

Grades breakdown:

Correct implementation and demo of “make grade” (or “cowtest” and/or “simplefork,” if not fully complete): 65%

As described in this web site (<https://www.rose-hulman.edu/class/csse/csse332/2425b/labs/cow/>), there are four main tasks and incremental results. If you do not complete the project fully, please let us know which tasks you completed (you believe) and generate the test output associated with that task in the above web page.

- Step 1 – Modifying fork Behavior: 20%
- Step 2 – Handling Segmentation Faults: 15%
- Step 3 – Adding reference counting: 15%
- Step 4 - Modifying copyout: 15%

Clear and detailed explanation of the changes made: 15%

Clear and detailed explanation of system call workflow: 15%