COMPUTER SCIENCE AND ENGINEERING
Indian Institute of Technology Palakkad
CS3110: Operating Systems Lab
Lab 7

30-Sep-2024

1. Implement a user-level program `pgtblprint` for xv6. When invoked, this program should print the kernel page table (valid pages only) in the following format. Note that xv6 for 64-bit RISC-V uses $Sv39$ addressing scheme.

   ```
   VA 0x0000000000000000: PT @ 0x0000000087f22000 -> PT @ 0x0000000087f1e000
   -> PT @ 0x0000000087f1d000 -> Page @ 0x0000000087f1f000
   VA 0x0000000000001000: PT @ 0x0000000087f22000 -> PT @ 0x0000000087f1e000
   -> PT @ 0x0000000087f1d000 -> Page @ 0x0000000087f1c000
   VA 0x0000000000002000: PT @ 0x0000000087f22000 -> PT @ 0x0000000087f1e000
   -> PT @ 0x0000000087f1d000 -> Page @ 0x0000000087f1b000
   VA 0xffffffffffffd000: PT @ 0x0000000087f22000 -> PT @ 0x0000000087f21000
   -> PT @ 0x0000000087f20000 -> Page @ 0x0000000087f20000
   VA 0xffffffffffffe000: PT @ 0x0000000087f22000 -> PT @ 0x0000000087f21000
   -> PT @ 0x0000000087f20000 -> Page @ 0x0000000087f40000
   ```

2. Implement a system call that takes two integer arguments x and y. When invoked, it should print page-table information, of the invoking process, from virtual page numbers x to $x + y$. A possible output is as follows.

   ```
   Virtual page number: 10
   Physical frame number: 20
   User accessible: No
   R|W|X: Yes|Yes|No

   Virtual page number: 11
   Physical frame number: Not available
   ```

3. Some operating systems (e.g., Linux) speed up certain system calls by sharing data in a read-only region between userspace and the kernel. Your task is to devise a safe-mechanism to obtain the PID in xv6 without switching to the kernel space. A possible way is to map one read-only page at `USYSCALL` (a virtual address) to the PCB when the process is created. For more hints visit `https://pdos.csail.mit.edu/6.828/2024/labs/pgtbl.html`.

4. Write C program(s) to understand the variation in hit-rate as a function of page cache size under OPT, FIFO, RAND, LRU, Approx. LRU page replacement policies and the following workloads: (a). no locality workload, (b). 80-20 workload, and (c). looping-sequential workload. Use a plotting utility to visualize the results. For more details refer Section 22.6 of `https://pages.cs.wisc.edu/~remzi/OSTEP/vm-beyondphys-policy.pdf`.