

Lab5Answers

To test the correctness, we develop several test cases, below is the test results when mixed cases is tested together, and the explanation is as follow

1.Null process: pid=0

The page directory of the null process is created after the creation of 5 global page tables, so it stays at the 6th frame in the 1000 kernel frames so the address will be $(1024+5) * 4096 = 42147864$. We could see below that pdir for null process (pid=0) is exactly that number. And this is no page fault.

2. Process generated by create ():

main process pid=1

Main process (pid=1) has its private page directory, at 4218880, which is the 7th frame in the 1000 kernel frame. Since it does not have virtual heap, there is no page fault.

If we use `vgertmem(2000)` in main we will get `SYSEERROR`

testm address -1

3. Process generated by vcreate();

Test1 pid 2: execute `char *add vgetmem(2000)` for 3 times.

Test 1 gets a private page directory different from null and main process. Since each page size is 4096, it will create a page table and map two page. It will generate 2 pages faults. After its execution, it will free the frame used (page, page table and directory)

pid is 2,pdir is 4222976

Page fault for address 16777220

Created page table 8

pagecount is 1, pid is 2// num of page in the page table

test1 address 16777216// virtual address

test1 address 16779216

Page fault for address 16783216

pagecount is 2, pid is 2

After sleepms (500)

Vcreate process Test 2 (pid3) which execute char *add= vgetmem(6000),

Test 2 also gets a private page directory from null and main process. The address is the same as test 1. This verified that the frame used by test 1 is freed after it terminates

It will also generate 2 page faults and map two virtual pages since 6000>4096.

ctxsw pid is 3,pdbr is 4222976

Page fault for address 16777220

Created page table 8 // frame id to store this page table

pagecount is 1, pid is 3

Page fault for address 16783216

pagecount is 2, pid is 3

test2 address 16777216

Deleted page table 8