CS450 Operating Systems (Spring 2018)

Josiah Hunt (jhunt5@hawk.iit.edu) (20350987)

Mayank Bansal (mbansal5@hawk.iit.edu) (20392482)

Programming Assignment 3

Part 2 (System call to display memory usage of a process)

TESTCASES

There are a total of 13 test programs that were written, all written incrementally complex.

0. Test case 0:

   Checking inital memory usage

   Use: to print the initial memory usage before the program starts.

   Result: shows initial usage of 3 pages (Code and Stack)

1. Test case 1:

   Adding 1024 bytes to memory

   Use: to check if a page is being allocated for this memory

   Result: shows usage of 4 pages (Code and Stack + 1 page)

2. Test case 2;

   Adding 4096 bytes to memory

Use: to check if a page is being allocated for this memory

Result: shows initial usage of 4 pages (Code and Stack + 1 page)

3. Test case 3:

   Adding 4096 bytes to memory
   Adding 4096 bytes to memory

   Use: to check if 2 pages are being allocated for this memory

   Result: shows usage of 5 pages (Code and Stack + 2 pages)

4. Test case 4:

   Adding 4097 bytes to memory

   Use: to check if 2 pages are being allocated for this memory

   Result: shows usage of 5 pages (Code and Stack + 2 pages)

5. Test case 5:

   Adding 4097 bytes to memory
   Removing 4097 bytes to memory

   Use: to check the initial memory is the same as the final memory

   Result: shows usage of 3 pages (Code and Stack + 0 pages)

6. Test case 6:

   Adding 56732 pages to memory

   Use: to check for large number of pages

   Result: shows usage of 65540 pages (Code and Stack + 65537 pages)

7. Test case 7:

   Adding 56750 pages to memory

   Use: to check if the process runs out of memory

Result: shows “out of memory” message after allocating as much as it could and then shows usage of 65540 pages (Code and Stack + 65537 pages)

8. Test case 8:

   Adding 1000 pages to memory
   Removing 1000 pages from memory

   Use: to check if init page count and final page count is the same after allotting pages.

   Result: shows usage of 3 pages (Code and Stack + 0 pages)

9. Test case 9:

   Adding 10 pages to memory of Parent Process
   Adding 10 pages to memory of Child Process

   Use: to check if the parent and child process use different process spaces and if we can allocate space independently

   Result: shows usage of 13 pages in parent, 13 pages in child (Code and Stack + 10 pages)

10. Test case 10:

    Adding 10 pages to memory of Parent Process
    Adding 5 pages to memory of Parent Process
    Adding 10 pages to memory of Child Process
    Removing 5 pages from memory of Child Process

    Use: to check if the parent and child process use different process spaces and if we can allocate space independently

    This validates Test Case 9

    Result: shows usage of 18 pages in parent, 8 pages in child (Code and Stack + 15 pages, 5 pages)

11. Test case 11:

    Adding 5 pages to memory of Parent Process
    Adding 10 pages to memory of Child Process
    Adding 15 pages to memory of Child of Child Process

    Use: to test if the process spaces work for multiple levels of processes. Here, there are 3 levels of process. Parent > Child > Child of Child

    Result: shows usage of 8 pages in parent, 13 pages in child, 18 pages in child of child (Code and Stack + 5 pages, 10 pages, 15 pages)

12. Test Case 12:

    Adding 56750 pages of memory in Parent process
    Adding 1000 pages of memory in Child Process

    Use: to check if the parent process can run out of memory

    Result: shows usage of 1003 pages in child, 56735 pages in parent (Code and Stack + 1000 pages, 56732 pages)