Ka Tam (A20374415)
Jack Critzer (A20396230)
CS 450
4/5/2020
Programming Assignment 3

Part 2 Test Results

myV2p(uint va, int w) System Call
1) The first test is on the address of the variables declared in the test program. This demonstrates the valid addresses in the program and proves that the function actually returns the physical address by giving the virtual address. As shown in the test data, both the virtual address and the physical address are off by 4 bytes from each other because a char is 4 bytes in size.
2) The second sets of test data are some random virtual addresses entered into the function. This demonstrates the error that will be returned if an invalid address is entered into the function. As shown in the test data, the function returns -1 because the addresses are all made up and they are not allocated in the memory.
3) The third set of test data is generated from the sbrk() function, which asks the operating system to generate a certain amount of heap space specified by the parameter. All those spaces are allocated, which means they are valid memory that can obtain the physical address from. This shows the virtual address translation without putting stuff in that memory slot.
4) The last set of test data is generated from the outside of the sbrk function. This set attempts to translate the virtual address outside of the sbrk allocation. This will result in invalid translation because those spaces are not present in the page table.

hasPages(int pid) System Call
1) The test program first calls hasPages() on the current process, which is the program itself. It initially has 1 page of data/text, 1 guard page, and 1 stack page. The heap pages have not been allocated yet.
2) Then, the process calls malloc, and allocates more pages in the heap, shown in the results of the second call to hasPages().
3) Next, the array is freed, and sbrk(10000) is called to increase the heap size, shown in the results of the third call.

```
This process, testHasPages, has the following pages:

Virtual Address  |  Permissions  |  Segment

----------------------------------------------

0          |      RWU       |    DATA/TEXT
1000       |      RW-       |    GUARD PAGE
2000       |      RWU       |    STACK
----------------------------------------------

This process, testHasPages, has the following pages:

Virtual Address  |  Permissions  |  Segment

----------------------------------------------

0          |      RWU       |    DATA/TEXT
1000       |      RW-       |    GUARD PAGE
2000       |      RWU       |    STACK
3000       |      RWU       |    HEAP
4000       |      RWU       |    HEAP
5000       |      RWU       |    HEAP
6000       |      RWU       |    HEAP
7000       |      RWU       |    HEAP
8000       |      RWU       |    HEAP
9000       |      RWU       |    HEAP
a000       |      RWU       |    HEAP
----------------------------------------------

This process, testHasPages, has the following pages:

Virtual Address  |  Permissions  |  Segment

----------------------------------------------

0          |      RWU       |    DATA/TEXT
1000       |      RW-       |    GUARD PAGE
2000       |      RWU       |    STACK
3000       |      RWU       |    HEAP
4000       |      RWU       |    HEAP
5000       |      RWU       |    HEAP
6000       |      RWU       |    HEAP
7000       |      RWU       |    HEAP
8000       |      RWU       |    HEAP
9000       |      RWU       |    HEAP
a000       |      RWU       |    HEAP
b000       |      RWU       |    HEAP
c000       |      RWU       |    HEAP
d000       |      RWU       |    HEAP
----------------------------------------------
```