Assignment 6 - Report

**Description**

In order to implement problem 9.1, we created function block\_space() which iterates through memlist and finds the number of available blocks.

To implement problem 9.4 we created three new functions: getmem\_new(), getstk\_new() and table().

To implement problem 9.7 we created a new function table2().

|  |  |
| --- | --- |
| **Question** | **Output** |
| 9.1 | xsh $ monitor 1  executing 9.1  The address of block 1 is 2164532992 and the size of the block  Max Stack position used in memory for process 4 till: 2684256104  Max memory space used for process is 4 37  is 519715072 |
| 9.4 | executing 9.4  Inside the main process having Process ID= 4  Before calling the child process , space statistics are :-  The address of block 1 is 2164532992 and the size of the block  custom made stack called for process id 4:  is 519715072  Inside the child process having Process ID= 5  The address of block 1 is 2164532992 and the size of the block  Max Stack position used in memory for process 5 till: 2684247944  Max memory space used for process is 5 29  Stack Memory not released for Process 5  Exiting the process having Process ID 5  is 519714048  custom made stack called for process id 4:  Inside the child process having Process ID= 6  The address of block 1 is 2164532992 and the size of the block  Max Stack position used in memory for process 6 till: 2684246920  Max memory space used for process is 6 29  Stack Memory not released for Process 6  Exiting the process having Process ID 6  is 519713024  Inside getmem\_new Heap Memory being allocated is 2048 to process 4  Before exiting the main process , final space statistics are :-  The address of block 1 is 2164535040 and the size of the block  Max Stack position used in memory for process 4 till: 2684255904  Max memory space used for process is 4 87  Stack Memory not released for Process 4  Exiting the process having Process ID 4  is 519710976 |
| 9.7 | xsh $ monitor 3  executing 9.7  Inside the main process having Process ID= 4  Before calling the process , space statistics are:-  The address of block 1 is 2164532992 and the size of the block  Max Stack position used in memory for process 5 till: 2684247992  Max memory space used for process is 5 17  is 519715072  Before exiting the main process , final space statistics are:-  The address of block 1 is 2164532992 and the size of the block  Max Stack position used in memory for process 4 till: 2684255904  Max memory space used for process is 4 87  is 519715072 |

**Implementation of 9.4**

As Xinu boots initially, there is only one contagious free block of memory and as new processes are created, memory is taken from this initial contagious block of free memory for the newly created processes. And when Kill() calls freestack() function, then the allotted chunk of memory is returned back. This process of allocation and de-allocation may results in creation of fragmented blocks of memory. However if the allocated blocks of memory are never returned then there will not be any fragmented blocks of free memory, only contagious memory will be there. Based on these concepts we have re-designed getmem() and getstk().

Since there are no fragments of free memory but only a contagious memory block, our getmem\_new() and getstk\_new() need not iterate through entire memlist . Now getmem\_new() just check if the available contagious block of memory is having enough space to satisfy the memory requirement and if enough memory is there then it scoops out memory from the lowest address space and return the base address . Similarly getstk\_new() check for the space and if space is available then it returns highest memory address as the base address .

**Implementation of 9.7**

To find how much memory on stack has been used by a process, we have initialized allotted memory blocks of the process with -99. Then we are recursively calling the process 5 times, this recursive call pushes local variable, return value, formal parameter and activation record on the process’s stack. So those memory blocks which holds these values will not contain -99. To know the amount of memory block used by the process, just before process exits, we count memory blocks not having -99 stored in them. This gives number of memory block used by the process.

This initialization of memory blocks with -99 is done in create() system call . Because initialization done at any other place does not get reflected.

**Functionality**

|  |  |  |
| --- | --- | --- |
| **File / Function** | **Description** | **User ID** |
|  |  |  |
| prototyes.h | Contains declaration of new system calls | pandeyh |
| process.h | A new member variable has been added to the structure procent here:  Int16 free\_flag  free\_flag is a variable that allows us to know whether memory is to be released for a particular process . It’s initialized to 1, and to 2 when memory has to be freed (2 has to explicitly set by calling the process).  They are for calling new getstk\_new() and to differ the call to freestk() | sshalabh |
| initialize.c | Here free\_flag is initialized to 1 | pandeyh |
| kill.c | Changes has been made to not call freestk for all those process whose free\_flag is initialized to 2 . | sshalabh |
| table.c | Contains code to test 9.4. This function gets its stack from getstk\_new() and thus its memory will not be de-allocated once it finishes execution . | sshalabh |
| table2.c | Contains code to test 9.7. This function gets its stack from getstk() and thus its memory will be de-allocated once it finishes execution .Also it shows the maximum stack memory usage of process . Here table2 is called recursively 5 times to push various values on to its stack and then remaining free block is calculated . Remaining free blocks contains value -99. | pandeyh |
| shell.c | Contains declaration for command monitor | pandeyh |
| monitor.c | This is the main file which integrates functionality implementing 9.1, 9.4 and 9.7. This file also has logic for Help. | sshalbh |
| create.c | It contains code to call getstk or getstck\_new depending on the value of free\_flag for implementing 9.4.Also has code to initialize stack memory blocks of a process with -99 for 9.7. | pandeyh |
| memmontr.h | Header file containing global variables and prototype declarations. | sshalabh |
| block\_space.c | This file has function which calculates number of free block of memory. Implementation of 9.1 | sshalabh |
| getstk\_new.c | New implementation of getstk() where memlist is not traversed . | sshalabh |
| getmem\_new.c | New implementation of getmem() where memlist is not traversed . | sshalabh |

**Tasks and responsibilities**

|  |  |
| --- | --- |
| **Tasks** | **UserID** |
| Unit Test Plan | Sshalabh/pandeyh |
| Testing | Sshalabh/pandeyh |
| Report | Sshalabh/pandeyh |