Operating Systems
CS-450
Assignment #3
(Group)

Sivasenthil Namachivayan
A20391478
Sravan Kumar Allu
A20343634

April 8,2017

Discussion on the implementation:

Changes done in the following files

sysproc.c

1. Declared variable pagecount, kfreecount, kalloccount as int datatype
2. The sizes of memory is found out using the system call sys_getmysize(void) where the variables a and b are declared where ‘a’ denotes the size of the address space consumed by the user program which was declared in the the file proc.h in xv6 OS and ‘b’ denotes the number of pages which was used by the user program. This will be like a/4096 will give the value of b.
3. The free memory pages are retried using the declared variable pagecount
4. The difference between the kfree (1 page is released when ever this is called) and kalloc(1 page is been allocated when ever this is called) calls

   Code Snippet:

```
int sys_getmysize(void)
{
        int a,b;
        a=proc→sz;// declared in the proc.h file in xv6
                cprintf(“\n\n The size of the address space is %d bytes\n”,a);
        b=a/4096;
                cprintf(“\n The number of pages is %d \n”,b);
        cprintf(“\n The free memory pages %d \n”, pagecount);
        cprintf(“\n The difference between kfree and kalloc calls %d \n”, kfreecount-
                kalloccount);
        return 0;
}
```

syscall.c

1. Declared the user defined

   variable getmysize[] as extern int array

   system call [SYS_getmysize] sys_getmysize

myprog.c – User Program

In this program, the user defined system call SYS_getmysize is invoked to get the memory usage, pages, free pages. In this user program I have included “malloc” to allocate the space for the program and also to get the maximum size that can be used from the operating system until it run out of memory.

Code Snippet:

```
int main(void)
{
        a=(chat *)malloc(10000);// this malloc value is been increased to get the
                                        //memory allocation error
        strcpy(a, “hello”);
        getmysize(); // system call
        exit();
}
```

vm.c

Under the function setupkvm(void) include the print statement to print the following values pagedirectory (pgdir), Phystop and kmap values.

Code Snippet:

```
cprintf("\n setupkvm: the pgdir value is %d, phystop is %d, kmap is %d\n",
        pgdir,PHYSTOP);
```

kalloc.c

Include the kfreecount and pagecount variable and do post increment when ever the function void kfree() is called. This will help us to check the number of times the memory is freed from the process.

Include the kalloccount variable and do the post increment when ever the function kalloc(void) is called to check the number of times the kalloc is used to allocate the memory for the process. Include variable pagecount-- so that the page will be reduced to identify the free list.

These variables are used in our user systemcall getmysize to get the desired value for this assignment.

syscall.h

In this file defined the user system call as SYS_getmysize with the index value of 23

user.h

In this file declared the system call as int getmysize(void);

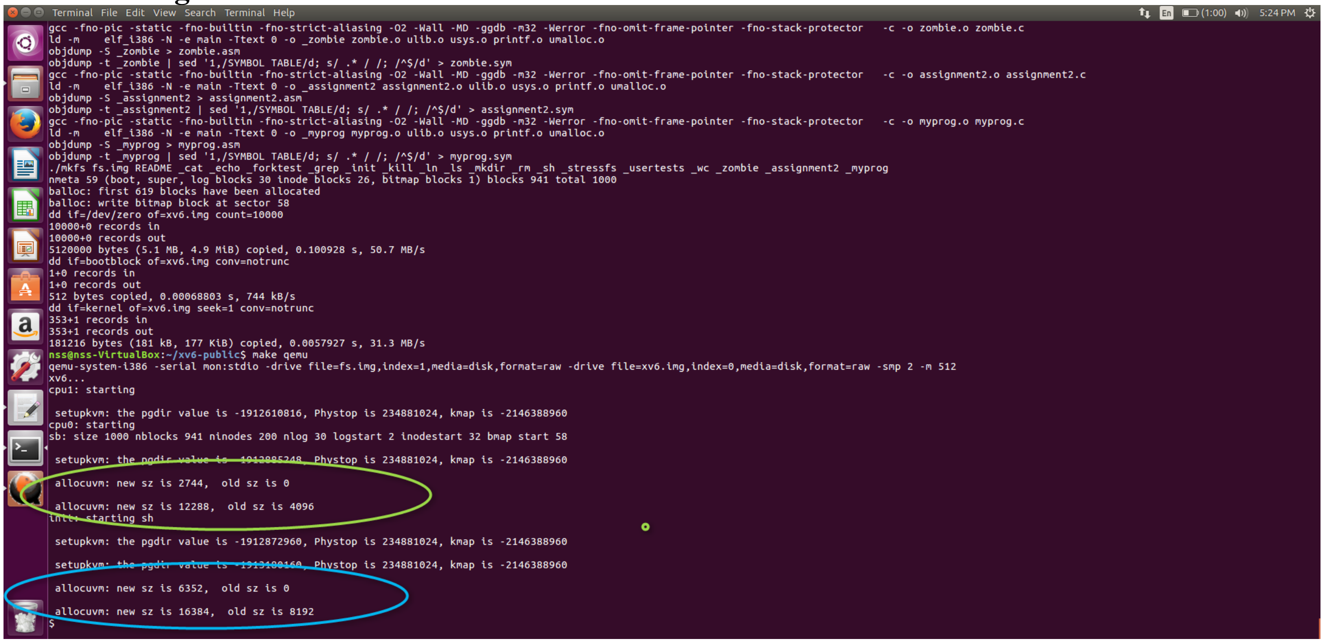
make

In this file mentioned the user program myprog.c under UPROGS=\ as _myprog for the user program to get executed by the xv6

Output:

With all the changes mentioned above, the user defined system call is invoked by the user program and the output is printed in the xv6 kernel.

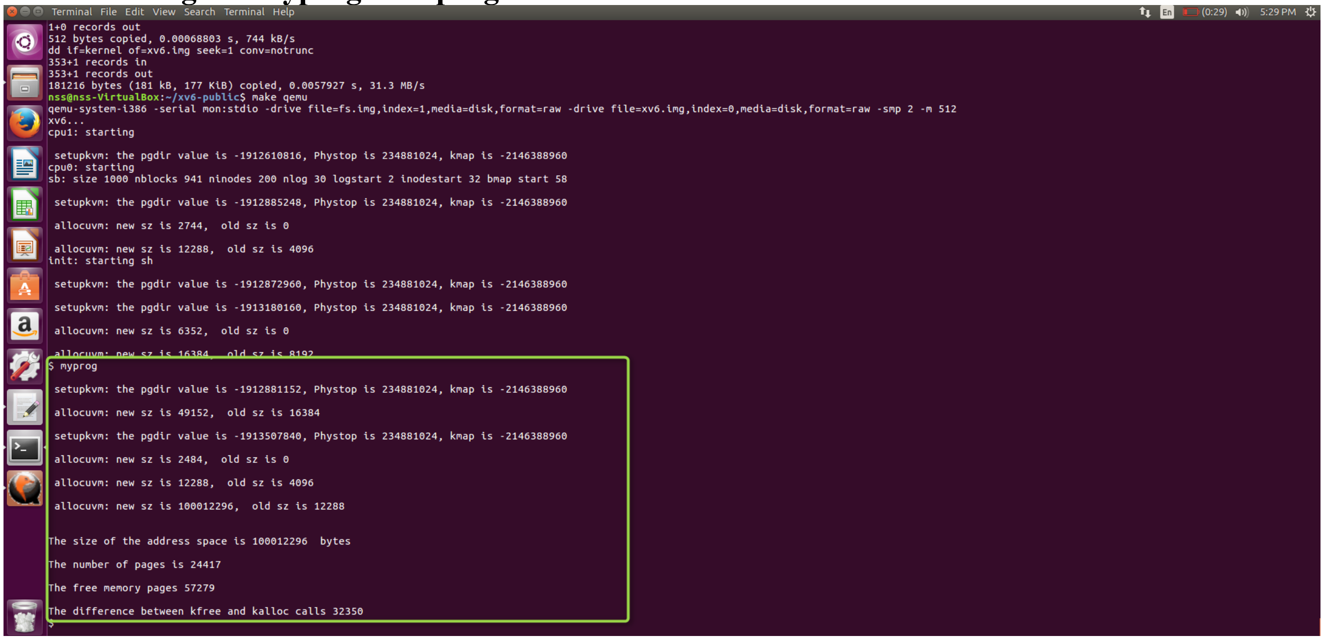
1. On booting xv6



```
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector   -c -o zombie.o zombie.c
ld -m    elf_i386 -N -e main -Ttext 0 -o _zombie zombie.o ulib.o usys.o printf.o umalloc.o
objdump -S _zombie > zombie.asm
objdump -t _zombie | sed '1,/SYMBOL TABLE/d; s/ .* / /; /^$/d' > zombie.sym
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector   -c -o assignment2.o assignment2.c
ld -m    elf_i386 -N -e main -Ttext 0 -o _assignment2 assignment2.o ulib.o usys.o printf.o umalloc.o
objdump -S _assignment2 > assignment2.asm
objdump -t _assignment2 | sed '1,/SYMBOL TABLE/d; s/ .* / /; /^$/d' > assignment2.sym
gcc -fno-pic -static -fno-builtin -fno-strict-aliasing -O2 -Wall -MD -ggdb -m32 -Werror -fno-omit-frame-pointer -fno-stack-protector   -c -o myprog.o myprog.c
ld -m    elf_i386 -N -e main -Ttext 0 -o _myprog myprog.o ulib.o usys.o printf.o umalloc.o
objdump -S _myprog > myprog.asm
objdump -t _myprog | sed '1,/SYMBOL TABLE/d; s/ .* / /; /^$/d' > myprog.sym
./mkfs fs.img README _cat _echo _forktest _grep _init _kill _ln _ls _mkdir _rm _sh _stressfs _usertests _wc _zombie _assignment2 _myprog
nmeta 59 (boot, super, log blocks 30 inode blocks 26, bitmap blocks 1) blocks 941 total 1000
balloc: first 619 blocks have been allocated
balloc: write bitmap block at sector 58
dd if=/dev/zero of=xv6.img count=10000
10000+0 records in
10000+0 records out
5120000 bytes (5.1 MB, 4.9 MiB) copied, 0.100928 s, 50.7 MB/s
dd if=bootblock of=xv6.img conv=notrunc
1+0 records in
1+0 records out
512 bytes copied, 0.00068803 s, 744 kB/s
dd if=kernel of=xv6.img seek=1 conv=notrunc
353+1 records in
353+1 records out
181216 bytes (181 kB, 177 KiB) copied, 0.0057927 s, 31.3 MB/s
nss@nss-VirtualBox:~/xv6-public$ make qemu
qemu-system-i386 -serial mon:stdio -drive file=fs.img,index=1,media=disk,format=raw -drive file=xv6.img,index=0,media=disk,format=raw -smp 2 -m 512
xv6...
cpu1: starting

 setupkvm: the pgdir value is -1912610816, Phystop is 234881024, kmap is -2146388960
cpu0: starting
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58

 setupkvm: the pgdir value is -1912885248, Phystop is 234881024, kmap is -2146388960

 allocuvm: new sz is 2744,  old sz is 0

 allocuvm: new sz is 12288,  old sz is 4096
init: starting sh

 setupkvm: the pgdir value is -1912872960, Phystop is 234881024, kmap is -2146388960

 setupkvm: the pgdir value is -1913180160, Phystop is 234881024, kmap is -2146388960

 allocuvm: new sz is 6352,  old sz is 0

 allocuvm: new sz is 16384,  old sz is 8192
$
```

By the time the xv6 os is booted the actual allocation of the virtual user memory is shown as new size and old size. The change in size of allocated user virtual memory is shown for the understanding. The total physical memory is also mentioned.

2. On running the myprog user program on xv6



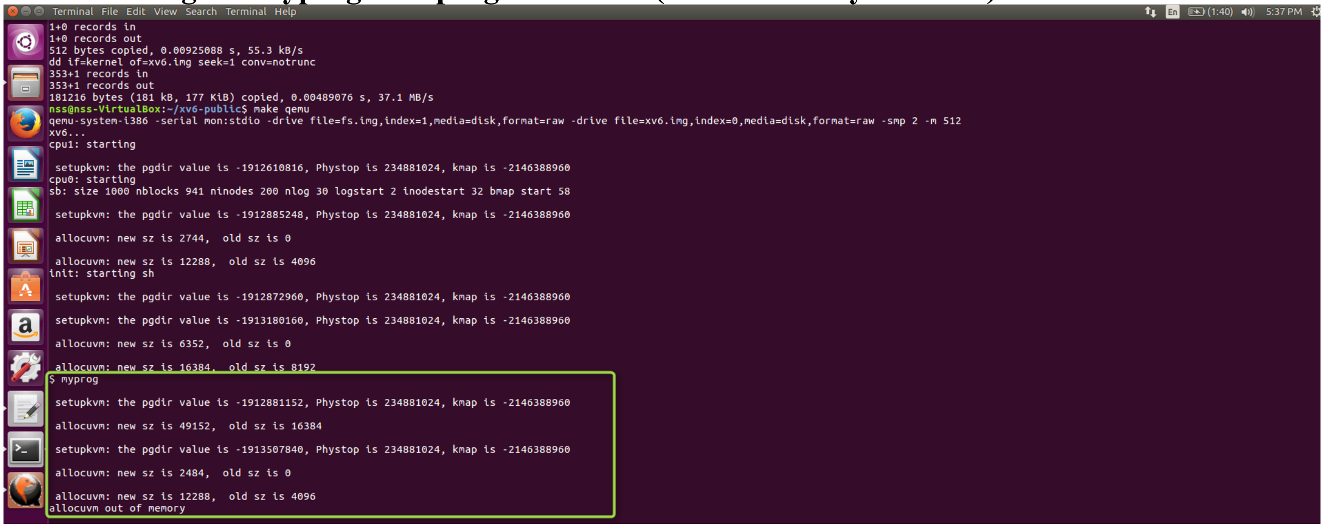
```
1+0 records out
512 bytes copied, 0.00068803 s, 744 kB/s
dd if=kernel of=xv6.img seek=1 conv=notrunc
353+1 records in
353+1 records out
181216 bytes (181 kB, 177 KiB) copied, 0.0057927 s, 31.3 MB/s
nss@nss-VirtualBox:~/xv6-public$ make qemu
qemu-system-i386 -serial mon:stdio -drive file=fs.img,index=1,media=disk,format=raw -drive file=xv6.img,index=0,media=disk,format=raw -smp 2 -m 512
xv6...
cpu1: starting

 setupkvm: the pgdir value is -1912610816, Phystop is 234881024, kmap is -2146388960
cpu0: starting
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58

 setupkvm: the pgdir value is -1912885248, Phystop is 234881024, kmap is -2146388960

 allocuvm: new sz is 2744,  old sz is 0

 allocuvm: new sz is 12288,  old sz is 4096
init: starting sh

 setupkvm: the pgdir value is -1912872960, Phystop is 234881024, kmap is -2146388960

 setupkvm: the pgdir value is -1913180160, Phystop is 234881024, kmap is -2146388960

 allocuvm: new sz is 6352,  old sz is 0

 allocuvm: new sz is 16384,  old sz is 8192
$ myprog

 setupkvm: the pgdir value is -1912881152, Phystop is 234881024, kmap is -2146388960

 allocuvm: new sz is 49152,  old sz is 16384

 setupkvm: the pgdir value is -1913507840, Phystop is 234881024, kmap is -2146388960

 allocuvm: new sz is 2484,  old sz is 0

 allocuvm: new sz is 12288,  old sz is 4096

 allocuvm: new sz is 100012296,  old sz is 12288


The size of the address space is 100012296  bytes

The number of pages is 24417

The free memory pages 57279

The difference between kfree and kalloc calls 32350
$
```

The user program runs and the size of the address space for the user program is 100012296 bytes and if we divide the total bytes by 4096 it will be 24417 pages which is also shown in the output.

As mentioned in point 1 the new size and the old size are changes as per the latest status.

The free memory pages available are 57279 and the difference between the number of times the kfree function and the kalloc functions called is 32350.

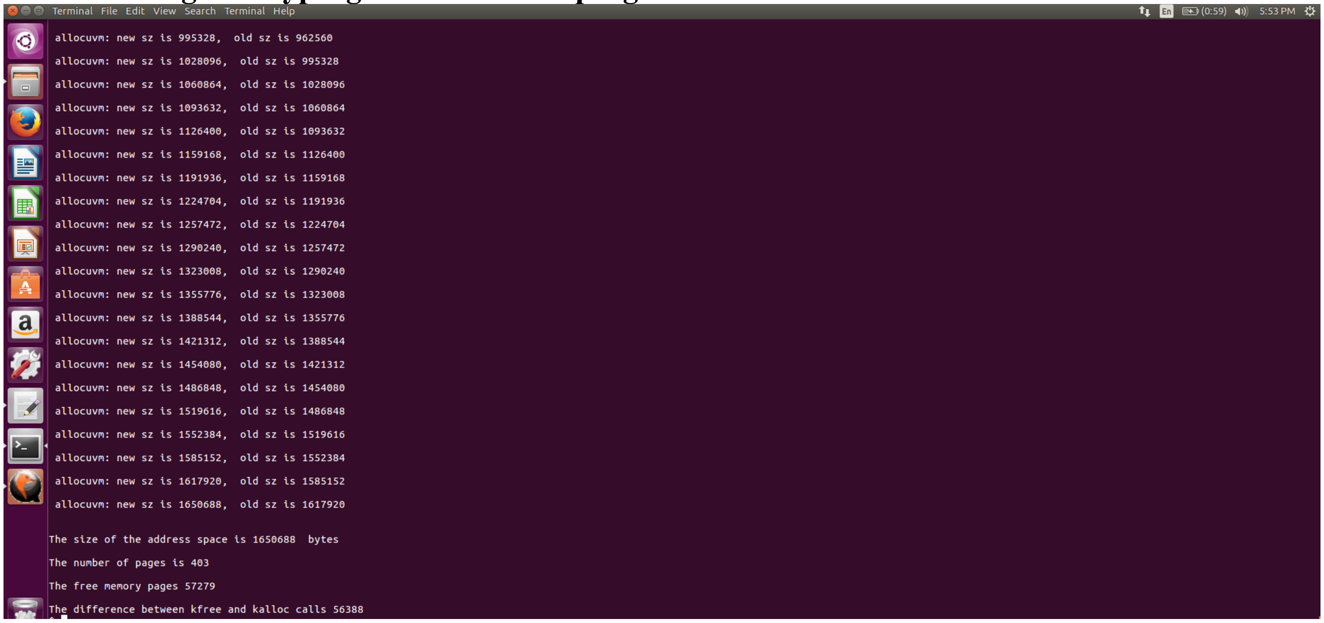
3. On running the myprog user program on xv6(out of memory scenario)



```
1+0 records in
1+0 records out
512 bytes copied, 0.00925088 s, 55.3 kB/s
dd if=kernel of=xv6.img seek=1 conv=notrunc
353+1 records in
353+1 records out
181216 bytes (181 kB, 177 KiB) copied, 0.00489076 s, 37.1 MB/s
nss@nss-VirtualBox:~/xv6-public$ make qemu
qemu-system-i386 -serial mon:stdio -drive file=fs.img,index=1,media=disk,format=raw -drive file=xv6.img,index=0,media=disk,format=raw -smp 2 -m 512
xv6...
cpu1: starting

 setupkvm: the pgdir value is -1912610816, Phystop is 234881024, kmap is -2146388960
cpu0: starting
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58

 setupkvm: the pgdir value is -1912885248, Phystop is 234881024, kmap is -2146388960

 allocuvm: new sz is 2744,  old sz is 0

 allocuvm: new sz is 12288,  old sz is 4096
init: starting sh

 setupkvm: the pgdir value is -1912872960, Phystop is 234881024, kmap is -2146388960

 setupkvm: the pgdir value is -1913180160, Phystop is 234881024, kmap is -2146388960

 allocuvm: new sz is 6352,  old sz is 0

 allocuvm: new sz is 16384,  old sz is 8192
$ myprog

 setupkvm: the pgdir value is -1912881152, Phystop is 234881024, kmap is -2146388960

 allocuvm: new sz is 49152,  old sz is 16384

 setupkvm: the pgdir value is -1913507840, Phystop is 234881024, kmap is -2146388960

 allocuvm: new sz is 2484,  old sz is 0

 allocuvm: new sz is 12288,  old sz is 4096
allocuvm out of memory
```

On trying to use the memory free memory using high value of malloc()(changed in the running user program and reworked), the memory bound error was thrown. So the test case is been handled until all the memory is consumed. The error shown in the output as allocuvm out of memory.

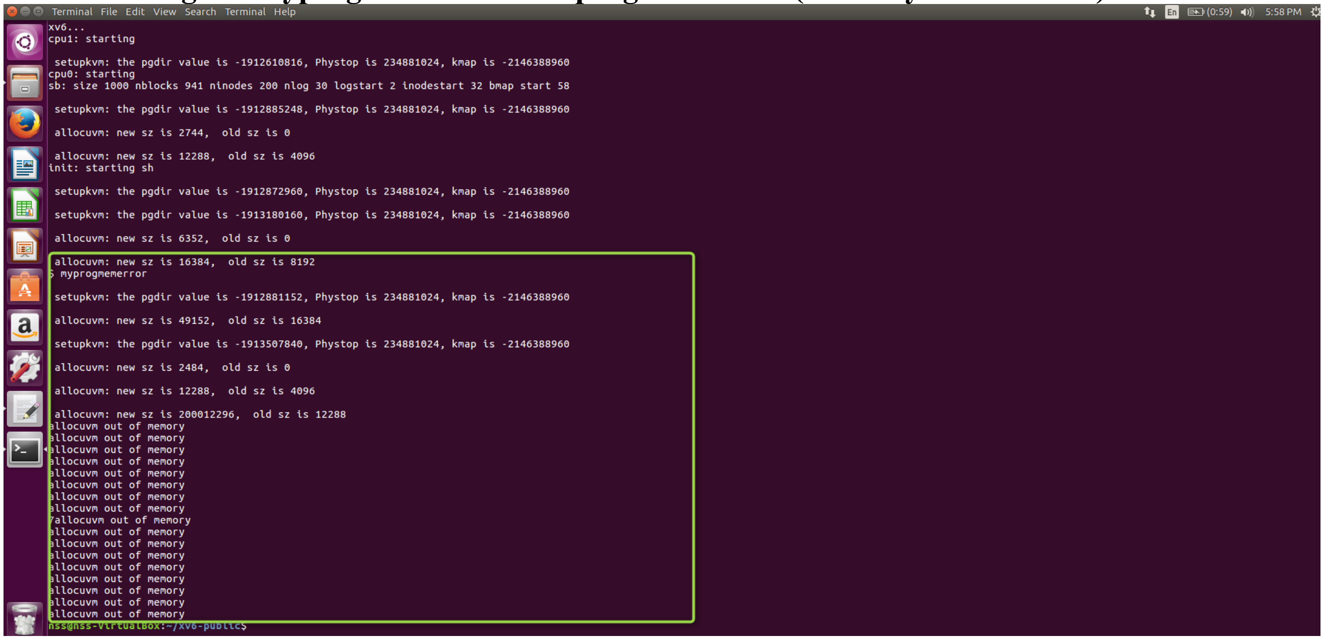
4. On running the myprogmemerror user program on xv6



```
 allocuvm: new sz is 995328,  old sz is 962560

 allocuvm: new sz is 1028096,  old sz is 995328

 allocuvm: new sz is 1060864,  old sz is 1028096

 allocuvm: new sz is 1093632,  old sz is 1060864

 allocuvm: new sz is 1126400,  old sz is 1093632

 allocuvm: new sz is 1159168,  old sz is 1126400

 allocuvm: new sz is 1191936,  old sz is 1159168

 allocuvm: new sz is 1224704,  old sz is 1191936

 allocuvm: new sz is 1257472,  old sz is 1224704

 allocuvm: new sz is 1290240,  old sz is 1257472

 allocuvm: new sz is 1323008,  old sz is 1290240

 allocuvm: new sz is 1355776,  old sz is 1323008

 allocuvm: new sz is 1388544,  old sz is 1355776

 allocuvm: new sz is 1421312,  old sz is 1388544

 allocuvm: new sz is 1454080,  old sz is 1421312

 allocuvm: new sz is 1486848,  old sz is 1454080

 allocuvm: new sz is 1519616,  old sz is 1486848

 allocuvm: new sz is 1552384,  old sz is 1519616

 allocuvm: new sz is 1585152,  old sz is 1552384

 allocuvm: new sz is 1617920,  old sz is 1585152

 allocuvm: new sz is 1650688,  old sz is 1617920


The size of the address space is 1650688  bytes

The number of pages is 403

The free memory pages 57279

The difference between kfree and kalloc calls 56388
```

on running the new user program – myprogmemerror executes the memory in loop when allocating the memory for the user program.

5. On running the myprogmemerror user program on xv6(Memory bound error)



```
xv6...
cpu1: starting

 setupkvm: the pgdir value is -1912610816, Phystop is 234881024, kmap is -2146388960
cpu0: starting
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58

 setupkvm: the pgdir value is -1912885248, Phystop is 234881024, kmap is -2146388960

 allocuvm: new sz is 2744,  old sz is 0

 allocuvm: new sz is 12288,  old sz is 4096
init: starting sh

 setupkvm: the pgdir value is -1912872960, Phystop is 234881024, kmap is -2146388960

 setupkvm: the pgdir value is -1913180160, Phystop is 234881024, kmap is -2146388960

 allocuvm: new sz is 6352,  old sz is 0

 allocuvm: new sz is 16384,  old sz is 8192
$ myprogmemerror

 setupkvm: the pgdir value is -1912881152, Phystop is 234881024, kmap is -2146388960

 allocuvm: new sz is 49152,  old sz is 16384

 setupkvm: the pgdir value is -1913507840, Phystop is 234881024, kmap is -2146388960

 allocuvm: new sz is 2484,  old sz is 0

 allocuvm: new sz is 12288,  old sz is 4096

 allocuvm: new sz is 200012296,  old sz is 12288
allocuvm out of memory
allocuvm out of memory
allocuvm out of memory
allocuvm out of memory
allocuvm out of memory
allocuvm out of memory
allocuvm out of memory
allocuvm out of memory
7allocuvm out of memory
allocuvm out of memory
allocuvm out of memory
allocuvm out of memory
allocuvm out of memory
allocuvm out of memory
allocuvm out of memory
allocuvm out of memory
allocuvm out of memory
nss@nss-VirtualBox:~/xv6-public$
```

On executing the user program myprogmemerror in loop for allocating the memory for the user program it allocates the initial available memory and then when all the memory is allocated it throws the error “allocuvm out of memory”.

README:

A new xv6 with modified kernel to support the changes done for assignment2-
introduce a new system call that returns the number of total system calls invoked.

To run the program:
1.Copy the rar file attached to the desired location in ubuntu OS
2.Extract the rar file
3.Navigate to the xv6 folder location in the ubuntu terminal using cd command
4.Type make clean in ubuntu terminal to clean the object files
5.Type make to compile the codes in ubuntu terminal.
6.Type make qemu to boot xv6 operating system in ununtu terminal.

Once the boot is done, you can mention the name of my C program which invoke the systemcall to print the output

7.Type myprog to run the user program- (shows all the memory details without error)
8.The output will show all the details as mentioned in the screenshot
9.Type again as myprogmemerror to run the loop user program(throw error when executed in loop)
10.You will notice that the program runs in loop and thrown the error.
11. Change the value of malloc() to higher value (1000000000)in myprog to check the error which is displayed in the **3. On running the myprog user program on xv6(out of memory scenario)**