Manual

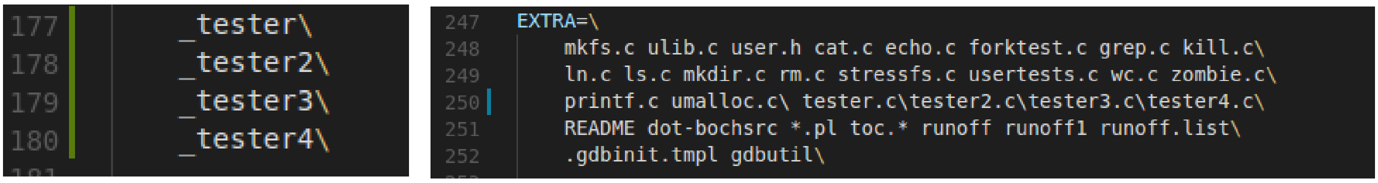
Description:

myMemory() prints the data obtained from the write user present bits in page table and page directory. It is also designed to print the total free pages available for user process and total pages used by the process.

Modifications

1)MakeFile

Created four test programs for specific cases - basic allocation, overallocation, allocation and free, and two process.



```
	_tester\
	_tester2\
	_tester3\
	_tester4\
```

```
EXTRA=\
	mkfs.c ulib.c user.h cat.c echo.c forktest.c grep.c kill.c\
	ln.c ls.c mkdir.c rm.c stressfs.c usertests.c wc.c zombie.c\
	printf.c umalloc.c\ tester.c\tester2.c\tester3.c\tester4.c\
	README dot-bochsrc *.pl toc.* runoff runoff1 runoff.list\
	.gdbinit.tmpl gdbutil\
```

Fig.1 adding the user programs in make file

2)syscall.h
Create the system call number for myMemory() system call. This can be found in line 23.

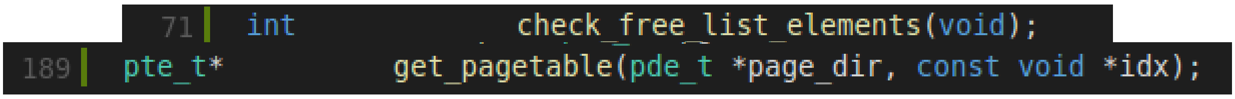
3)syscall.c
Extern declaration for myMemory() system call and addition of it in syscalls array.
Lines: 106,130.

4) user.h
Declaration of myMemory system call
Lines: 26.

5) Usys.S

Insert the new system call here too like we did in PA2
Lines: 32.

6)Defs.h
Define a function that would be helpful in counting number of free pages and also a function to return the page table info for a directory entry.

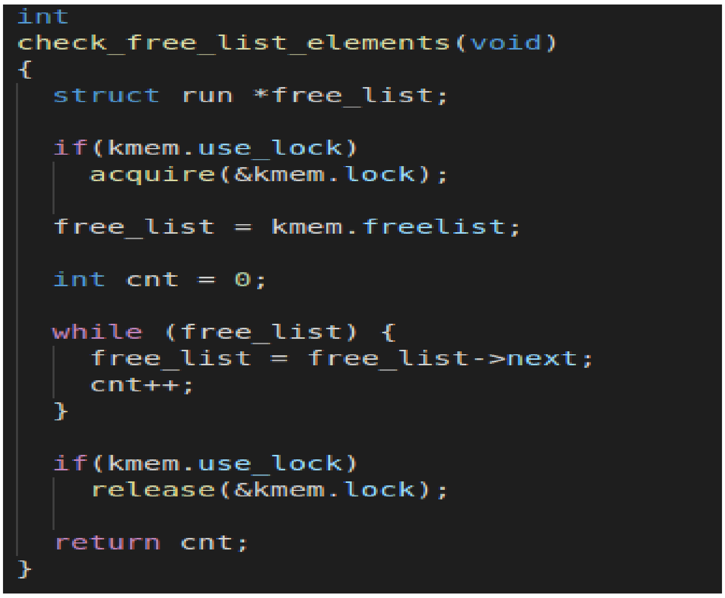


```
int             check_free_list_elements(void);
pte_t*          get_pagetable(pde_t *page_dir, const void *idx);
```

Fig.2 Added function declaration in Defs.h

7) kalloc.c

check_free_list_elements()
The idea is to check thru kmem.freelist and return the number of free pages

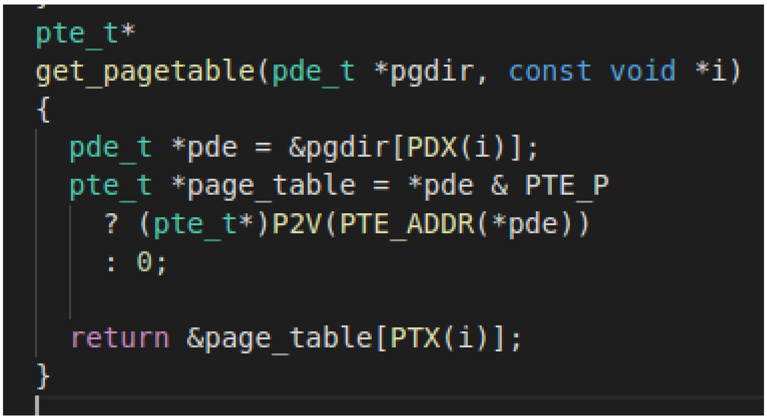


```
int
check_free_list_elements(void)
{
  struct run *free_list;

  if(kmem.use_lock)
    acquire(&kmem.lock);

  free_list = kmem.freelist;

  int cnt = 0;

  while (free_list) {
    free_list = free_list->next;
    cnt++;
  }

  if(kmem.use_lock)
    release(&kmem.lock);

  return cnt;
}
```

Fig.3 The function in kalloc.c

8) vm.c

Here, we define the get_pagetable function which returns page table entry address from the passed page directory entry.



```
pte_t*
get_pagetable(pde_t *pgdir, const void *i)
{
  pde_t *pde = &pgdir[PDX(i)];
  pte_t *page_table = *pde & PTE_P
    ? (pte_t*)P2V(PTE_ADDR(*pde))
    : 0;

  return &page_table[PTX(i)];
}
```

Fig.4 The function in vm.c

9) Types.h

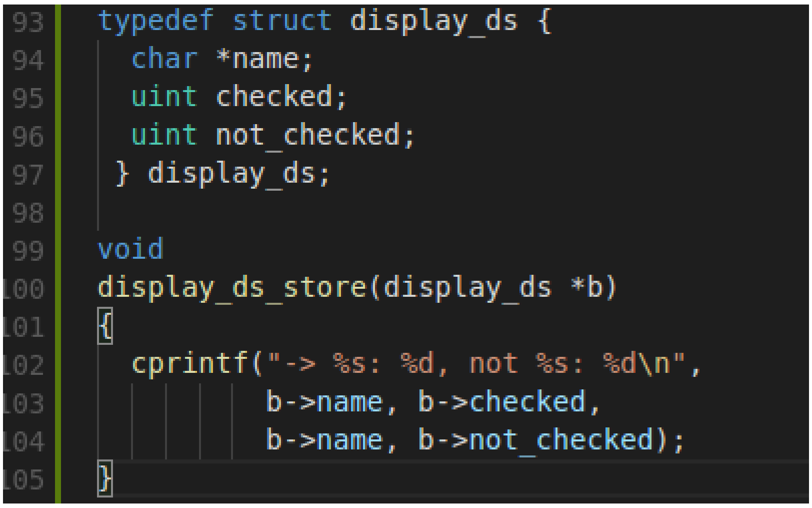
We add another macro similar to the already existing pde_t.

Line; 5.

10) sysproc.c

The definition of the system call is done here.

We traverse thru proc-> pgdir and we get the page table entry thru get_pagetable and log the entries with PTE_P (Present) PTE_U (User) PTE_W (writable). Then, we get the free pages by calling check_free_elements. For display purpose we define a datastructure and a function. The datastructure display_ds stores the information of each object (user, present and writable). The function display_ds_store(display_ds *b) displays the object in a simple format.



```
93   typedef struct display_ds {
94     char *name;
95     uint checked;
96     uint not_checked;
97    } display_ds;
98   
99   void
100  display_ds_store(display_ds *b)
101  {
102    cprintf("-> %s: %d, not %s: %d\n",
103            b->name, b->checked,
104            b->name, b->not_checked);
105  }
```

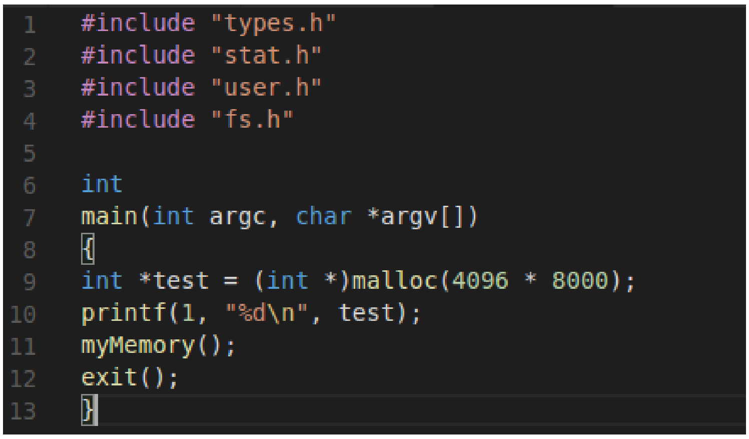
Fig. 5 The datastrucute and the function

modified Code lines: 93 to 177

Test programs

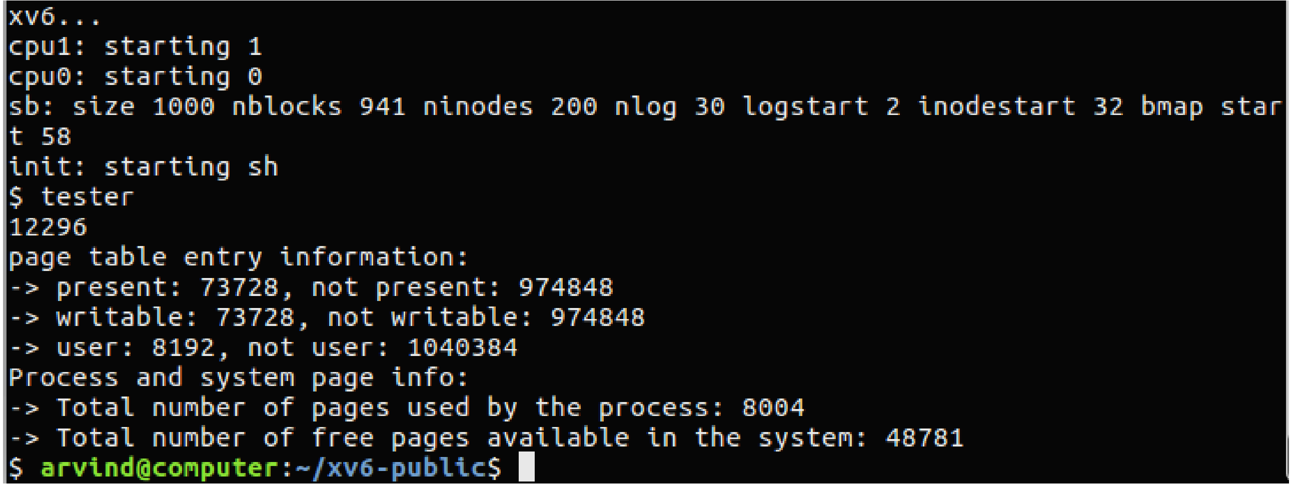
1) Basic allocation

tester.c contains basic declaration of 8000 pages.
Code:



```
#include "types.h"
#include "stat.h"
#include "user.h"
#include "fs.h"

int
main(int argc, char *argv[])
{
int *test = (int *)malloc(4096 * 8000);
printf(1, "%d\n", test);
myMemory();
exit();
}
```

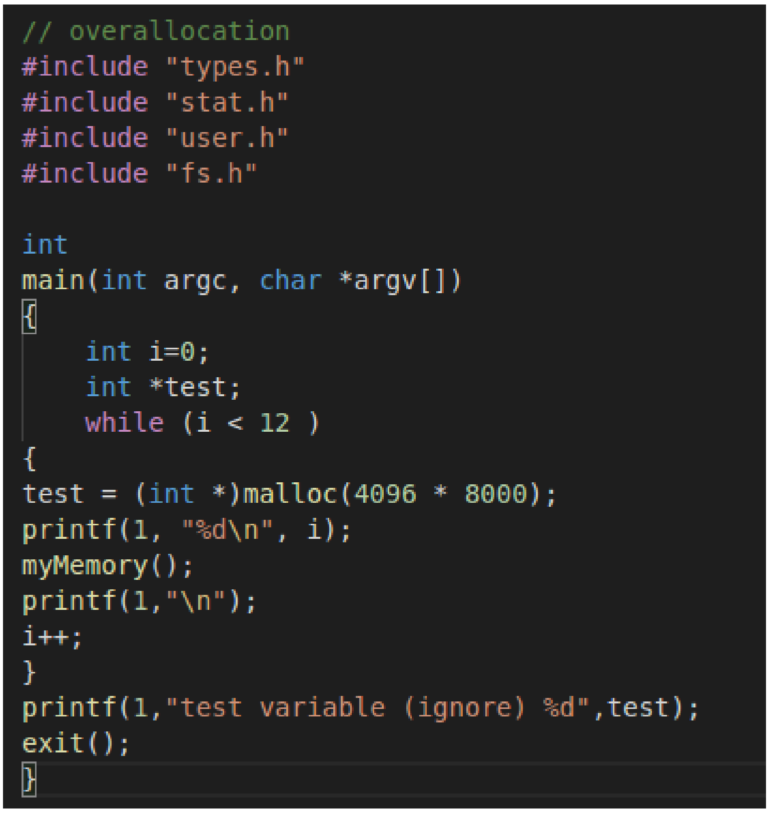
Output:



```
xv6...
cpu1: starting 1
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap star
t 58
init: starting sh
$ tester
12296
page table entry information:
-> present: 73728, not present: 974848
-> writable: 73728, not writable: 974848
-> user: 8192, not user: 1040384
Process and system page info:
-> Total number of pages used by the process: 8004
-> Total number of free pages available in the system: 48781
$ arvind@computer:~/xv6-public$ 
```

2) Overallocation

From the info we could see that there are 48781 free pages left. Now we intend to allocate more than this requirement.
Code:



```
// overallocation
#include "types.h"
#include "stat.h"
#include "user.h"
#include "fs.h"

int
main(int argc, char *argv[])
{
    int i=0;
    int *test;
    while (i < 12 )
{
test = (int *)malloc(4096 * 8000);
printf(1, "%d\n", i);
myMemory();
printf(1,"\n");
i++;
}
printf(1,"test variable (ignore) %d",test);
exit();
}
```

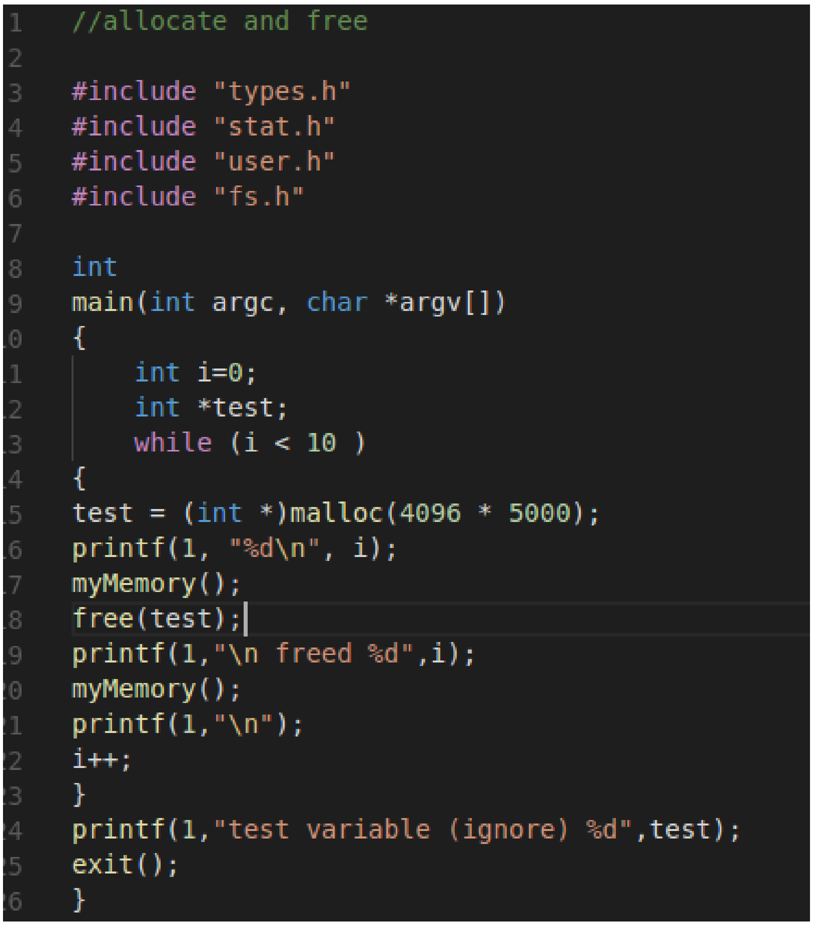
Output

```
xv6...
cpu1: starting 1
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
$ tester2
0
page table entry information:
-> present: 73728, not present: 974848
-> writable: 73728, not writable: 974848
-> user: 8192, not user: 1040384
Process and system page info:
-> Total number of pages used by the process: 8004
-> Total number of free pages available in the system: 48781

1
page table entry information:
-> present: 81920, not present: 966656
-> writable: 81920, not writable: 966656
-> user: 16384, not user: 1032192
Process and system page info:
-> Total number of pages used by the process: 16004
-> Total number of free pages available in the system: 40773

2
page table entry information:
-> present: 90112, not present: 958464
-> writable: 90112, not writable: 958464
-> user: 24576, not user: 1024000
Process and system page info:
-> Total number of pages used by the process: 24004
-> Total number of free pages available in the system: 32765

3
page table entry information:
-> present: 98304, not present: 950272
-> writable: 98304, not writable: 950272
-> user: 32768, not user: 1015808
Process and system page info:
-> Total number of pages used by the process: 32004
-> Total number of free pages available in the system: 24757

4
page table entry information:
-> present: 106496, not present: 942080
-> writable: 106496, not writable: 942080
-> user: 40960, not user: 1007616
Process and system page info:
-> Total number of pages used by the process: 40004
-> Total number of free pages available in the system: 16749

5
page table entry information:
-> present: 113664, not present: 934912
-> writable: 113664, not writable: 934912
-> user: 48128, not user: 1000448
Process and system page info:
-> Total number of pages used by the process: 48004
-> Total number of free pages available in the system: 8742

6
page table entry information:
-> present: 121856, not present: 926720
-> writable: 121856, not writable: 926720
-> user: 56320, not user: 992256
Process and system page info:
-> Total number of pages used by the process: 56004
-> Total number of free pages available in the system: 734

allocuvm out of memory
7
page table entry information:
-> present: 121856, not present: 926720
-> writable: 121856, not writable: 926720
-> user: 56320, not user: 992256
Process and system page info:
-> Total number of pages used by the process: 56004
-> Total number of free pages available in the system: 733

allocuvm out of memory
8
page table entry information:
-> present: 121856, not present: 926720
-> writable: 121856, not writable: 926720
-> user: 56320, not user: 992256
Process and system page info:
-> Total number of pages used by the process: 56004
-> Total number of free pages available in the system: 733

allocuvm out of memory
9
page table entry information:
-> present: 121856, not present: 926720
-> writable: 121856, not writable: 926720
-> user: 56320, not user: 992256
Process and system page info:
-> Total number of pages used by the process: 56004
-> Total number of free pages available in the system: 733

allocuvm out of memory
10
page table entry information:
-> present: 121856, not present: 926720
-> writable: 121856, not writable: 926720
-> user: 56320, not user: 992256
Process and system page info:
-> Total number of pages used by the process: 56004
-> Total number of free pages available in the system: 733

allocuvm out of memory
11
page table entry information:
-> present: 121856, not present: 926720
-> writable: 121856, not writable: 926720
-> user: 56320, not user: 992256
Process and system page info:
-> Total number of pages used by the process: 56004
-> Total number of free pages available in the system: 733

test variable (ignore) 0$
```

3) Allocate and Free

This test case will allocate and free 5000 pages for 10 times. The results and the code is given below.

Code:



```
//allocate and free

#include "types.h"
#include "stat.h"
#include "user.h"
#include "fs.h"

int
main(int argc, char *argv[])
{
    int i=0;
    int *test;
    while (i < 10 )
{
test = (int *)malloc(4096 * 5000);
printf(1, "%d\n", i);
myMemory();
free(test);
printf(1,"\n freed %d",i);
myMemory();
printf(1,"\n");
i++;
}
printf(1,"test variable (ignore) %d",test);
exit();
}
```

Output:

```
xv6...
cpu1: starting 1
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
$ tester3
0
page table entry information:
-> present: 70656, not present: 977920
-> writable: 70656, not writable: 977920
-> user: 5120, not user: 1043456
Process and system page info:
-> Total number of pages used by the process: 5004
-> Total number of free pages available in the system: 51784

 freed 0page table entry information:
-> present: 70656, not present: 977920
-> writable: 70656, not writable: 977920
-> user: 5120, not user: 1043456
Process and system page info:
-> Total number of pages used by the process: 5004
-> Total number of free pages available in the system: 51784
```

```
1
page table entry information:
-> present: 70656, not present: 977920
-> writable: 70656, not writable: 977920
-> user: 5120, not user: 1043456
Process and system page info:
-> Total number of pages used by the process: 5004
-> Total number of free pages available in the system: 51784

 freed 1page table entry information:
-> present: 70656, not present: 977920
-> writable: 70656, not writable: 977920
-> user: 5120, not user: 1043456
Process and system page info:
-> Total number of pages used by the process: 5004
-> Total number of free pages available in the system: 51784

2
page table entry information:
-> present: 70656, not present: 977920
-> writable: 70656, not writable: 977920
-> user: 5120, not user: 1043456
Process and system page info:
-> Total number of pages used by the process: 5004
-> Total number of free pages available in the system: 51784

 freed 2page table entry information:
-> present: 70656, not present: 977920
-> writable: 70656, not writable: 977920
-> user: 5120, not user: 1043456
Process and system page info:
-> Total number of pages used by the process: 5004
-> Total number of free pages available in the system: 51784

3
page table entry information:
-> present: 70656, not present: 977920
-> writable: 70656, not writable: 977920
-> user: 5120, not user: 1043456
Process and system page info:
-> Total number of pages used by the process: 5004
-> Total number of free pages available in the system: 51784

 freed 3page table entry information:
-> present: 70656, not present: 977920
-> writable: 70656, not writable: 977920
-> user: 5120, not user: 1043456
Process and system page info:
-> Total number of pages used by the process: 5004
-> Total number of free pages available in the system: 51784
```

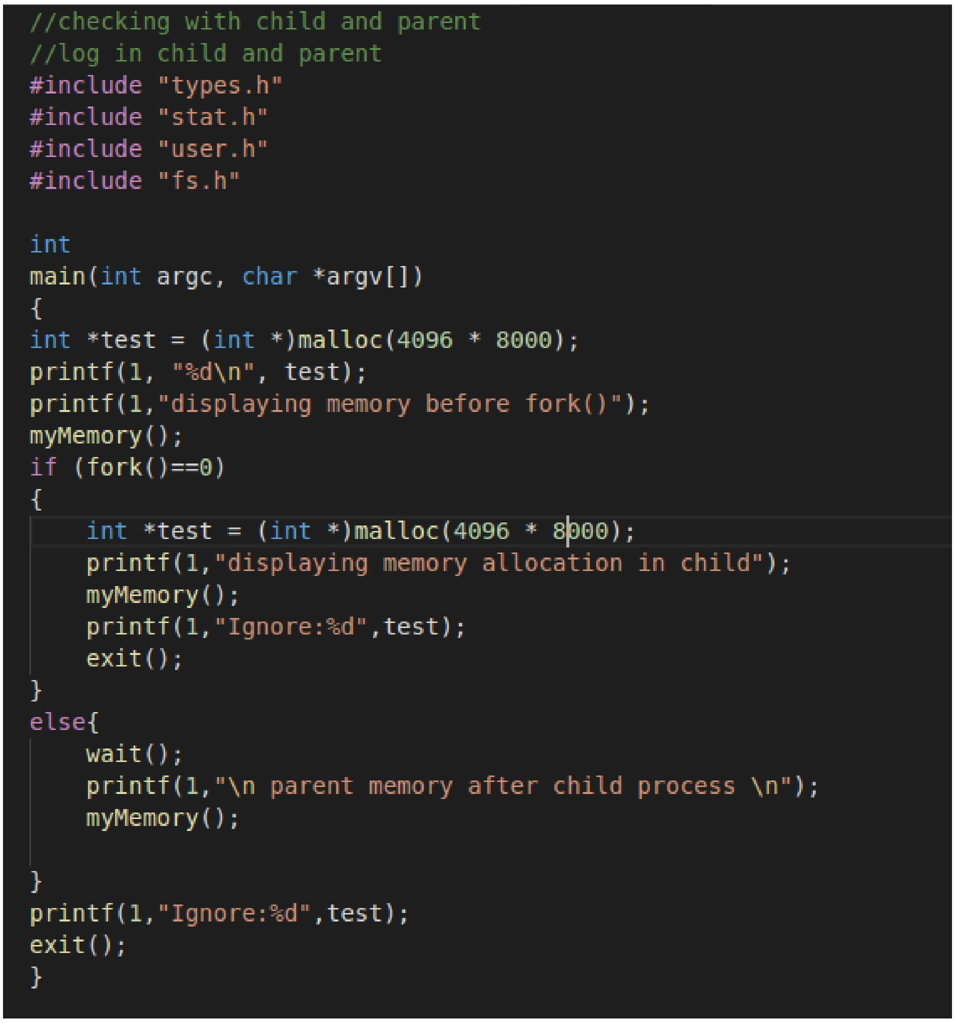
```
4
page table entry information:
-> present: 70656, not present: 977920
-> writable: 70656, not writable: 977920
-> user: 5120, not user: 1043456
Process and system page info:
-> Total number of pages used by the process: 5004
-> Total number of free pages available in the system: 51784

 freed 4page table entry information:
-> present: 70656, not present: 977920
-> writable: 70656, not writable: 977920
-> user: 5120, not user: 1043456
Process and system page info:
-> Total number of pages used by the process: 5004
-> Total number of free pages available in the system: 51784

5
page table entry information:
-> present: 70656, not present: 977920
-> writable: 70656, not writable: 977920
-> user: 5120, not user: 1043456
Process and system page info:
-> Total number of pages used by the process: 5004
-> Total number of free pages available in the system: 51784

 freed 5page table entry information:
-> present: 70656, not present: 977920
-> writable: 70656, not writable: 977920
-> user: 5120, not user: 1043456
Process and system page info:
-> Total number of pages used by the process: 5004
-> Total number of free pages available in the system: 51784

6
page table entry information:
-> present: 70656, not present: 977920
-> writable: 70656, not writable: 977920
-> user: 5120, not user: 1043456
Process and system page info:
-> Total number of pages used by the process: 5004
-> Total number of free pages available in the system: 51784

 freed 6page table entry information:
-> present: 70656, not present: 977920
-> writable: 70656, not writable: 977920
-> user: 5120, not user: 1043456
Process and system page info:
-> Total number of pages used by the process: 5004
-> Total number of free pages available in the system: 51784

7
```

```
page table entry information:
-> present: 70656, not present: 977920
-> writable: 70656, not writable: 977920
-> user: 5120, not user: 1043456
Process and system page info:
-> Total number of pages used by the process: 5004
-> Total number of free pages available in the system: 51784

 freed 7page table entry information:
-> present: 70656, not present: 977920
-> writable: 70656, not writable: 977920
-> user: 5120, not user: 1043456
Process and system page info:
-> Total number of pages used by the process: 5004
-> Total number of free pages available in the system: 51784

8
page table entry information:
-> present: 70656, not present: 977920
-> writable: 70656, not writable: 977920
-> user: 5120, not user: 1043456
Process and system page info:
-> Total number of pages used by the process: 5004
-> Total number of free pages available in the system: 51784

 freed 8page table entry information:
-> present: 70656, not present: 977920
-> writable: 70656, not writable: 977920
-> user: 5120, not user: 1043456
Process and system page info:
-> Total number of pages used by the process: 5004
-> Total number of free pages available in the system: 51784

9
page table entry information:
-> present: 70656, not present: 977920
-> writable: 70656, not writable: 977920
-> user: 5120, not user: 1043456
Process and system page info:
-> Total number of pages used by the process: 5004
-> Total number of free pages available in the system: 51784

 freed 9page table entry information:
-> present: 70656, not present: 977920
-> writable: 70656, not writable: 977920
-> user: 5120, not user: 1043456
Process and system page info:
-> Total number of pages used by the process: 5004
-> Total number of free pages available in the system: 51784

test variable (ignore) 12296$
```

4) The parent and child

We will allocate some memory for parent and then we will display memory details. We would also create a child and then allocate some memory there and check its memory details. Finally we would check the memory details after child completes.

Code:



```
//checking with child and parent
//log in child and parent
#include "types.h"
#include "stat.h"
#include "user.h"
#include "fs.h"

int
main(int argc, char *argv[])
{
int *test = (int *)malloc(4096 * 8000);
printf(1, "%d\n", test);
printf(1,"displaying memory before fork()");
myMemory();
if (fork()==0)
{
    int *test = (int *)malloc(4096 * 8000);
    printf(1,"displaying memory allocation in child");
    myMemory();
    printf(1,"Ignore:%d",test);
    exit();
}
else{
    wait();
    printf(1,"\n parent memory after child process \n");
    myMemory();

}
printf(1,"Ignore:%d",test);
exit();
}
```

Output:

```
xv6...
cpu1: starting 1
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
$ tester4
12296
displaying memory before fork()page table entry information:
-> present: 73728, not present: 974848
-> writable: 73728, not writable: 974848
-> user: 8192, not user: 1040384
Process and system page info:
-> Total number of pages used by the process: 8004
-> Total number of free pages available in the system: 48781
```

```
displaying memory allocation in childpage table entry information:
-> present: 81920, not present: 966656
-> writable: 81920, not writable: 966656
-> user: 16384, not user: 1032192
Process and system page info:
-> Total number of pages used by the process: 16004
-> Total number of free pages available in the system: 32695
Ignore:32780304
 parent memory after child process
page table entry information:
-> present: 73728, not present: 974848
-> writable: 73728, not writable: 974848
-> user: 8192, not user: 1040384
Process and system page info:
-> Total number of pages used by the process: 8004
-> Total number of free pages available in the system: 48781
Ignore:12296$
```