Parth Mangrola
861286610
Lab 21

Lab 3 Report

defs.h
- Modified copyuvm

memlayout.h
- Added USERSTACKTOP variable

proc,c

```
    if((np->pgdir = copyuvm(curproc->pgdir, curproc->sz, curproc->numPages)) == 0){ //lab3
```

- Modified copyuvm

proc.h
- Added numpages variable

trap.c
- Added page fault when the OS kills a process

```
    case T_PGFLT: ;
     uint addr = rcr2();
     uint pages = myproc()->numPages + 1;

    if(addr >= USERSTACKTOP - ((PGSIZE*pages) + 1)){
         if(allocuvm(myproc()->pgdir,PGROUNDDOWN(addr),PGROUNDDOWN(addr) + 1) == 0){
                 cprintf("case T_PGFLT from trap.c: allocuvm failed. Number of current allocated pages: %d\n", myproc()->numPages);
                 exit();
         }
         myproc()->numPages++;
         cprintf("case T_PGFLT from trap.c: allocuvm succeeded. Number of pages allocated: %d\n", myproc()->numPages);
         break;
    }
```

vm.c
- Modified copyuvm to copy memory for child process

```
for(i = USERSTACKTOP - PGSIZE + 1; spages > 0; i -= PGSIZE, spages--){ //lab3
  if((pte = walkpgdir(pgdir, (void *) i, 0)) == 0)
    panic("copyuvm: pte should exist");
  if(!(*pte & PTE_P))
    panic("copyuvm: page not present");
  pa = PTE_ADDR(*pte);
  flags = PTE_FLAGS(*pte);
  if((mem = kalloc()) == 0)
    goto bad;
  memmove(mem, (char*)P2V(pa), PGSIZE);
  if(mappages(d, (void*)i, PGSIZE, V2P(mem), flags) < 0)
    goto bad;
}
```

exec.c

- Allocated two pages, first is inaccessible and second is user stack
- KERNBASE is top of user memory
- USERSTACKTOP is one below KERNBASE
  - We are moving down now instead of up

```c
  uint stackSize = KERNBASE-PGSIZE;

  sz = PGROUNDUP(sz);
  if((allocuvm(pgdir, stackSize, stackSize+1)) == 0)
    goto bad;
   //clearpteu(pgdir, (char*)(sz - 2*PGSIZE));
  sp = USERSTACKTOP;
```

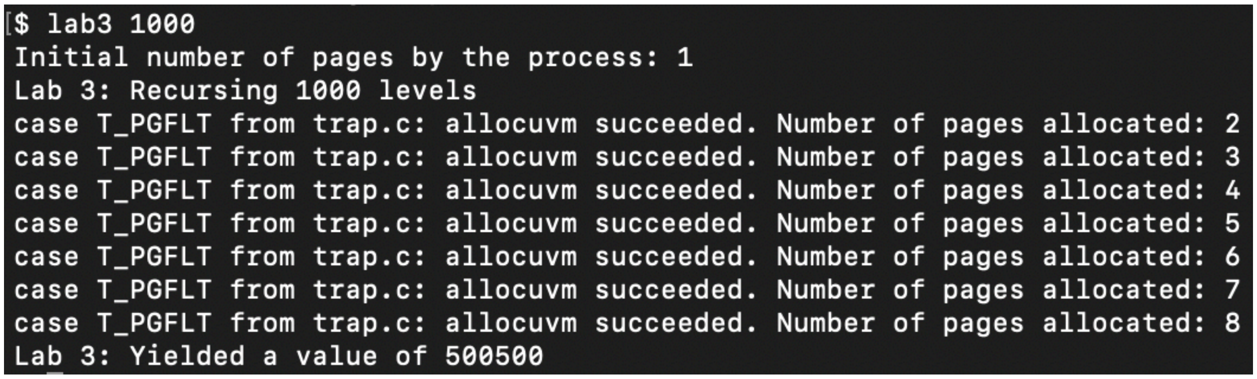
syscall.c

- Replaced all curproc-sz with USERSTACKTOP so address doesn’t go over it

```c
int
fetchint(uint addr, int *ip)
{
  // struct proc *curproc = myproc();

  if(addr >= USERSTACKTOP|| addr+4 > USERSTACKTOP)
    return -1;
  *ip = *(int*)(addr);
  return 0;
}

```



```
$ lab3 1000
Initial number of pages by the process: 1
Lab 3: Recursing 1000 levels
case T_PGFLT from trap.c: allocuvm succeeded. Number of pages allocated: 2
case T_PGFLT from trap.c: allocuvm succeeded. Number of pages allocated: 3
case T_PGFLT from trap.c: allocuvm succeeded. Number of pages allocated: 4
case T_PGFLT from trap.c: allocuvm succeeded. Number of pages allocated: 5
case T_PGFLT from trap.c: allocuvm succeeded. Number of pages allocated: 6
case T_PGFLT from trap.c: allocuvm succeeded. Number of pages allocated: 7
case T_PGFLT from trap.c: allocuvm succeeded. Number of pages allocated: 8
Lab 3: Yielded a value of 500500
```