

CIS-657 Spring-2019
Principals of Operating System

Programming Assignment-2

Divya Sai Sekhar, Mullapudi

SUID: 755658447

CIS657 Spring 2019

Assignment Disclosure Form

Assignment #:2

Name: Divya Sai Sekhar, Mullapudi

SUID :755658447

1. Did you consult with anyone other than instructor or TA/grader on parts of this assignment?

If Yes, please give the details.

No,

2. Did you consult an outside source such as an Internet forum or a book on parts of this assignment?

If Yes, please give the details.

No

I assert that, to the best of my knowledge, the information on this sheet is true.

Signature: _Divya Sai Sekhar Mullapudi_

Date : 04/14/2019

Design:

Task-1:

Read System Call:

Read system call is implemented by reading a char* from a string that was created in UNIX environment to char* that was created in user program space by user in nachos environment. This was done by copying the string to memory location of nachos main memory. After completion of copying PC of CPU is incremented.

Write System Call:

Write system call is implemented by writing a char* created by user program to output. It is done by read the char* created by user prog from main memory and printing it output. After completion of copying PC of CPU is incremented.

Fork System Call:

ThreadFork() is made by implementing by calling instantiating a new call new thread and passing function from kernel space as an argument to create a new thread space and incrementing the PC.

Yield System Call:

ThreadYield() is implemented by storing the incrementing the PC and yield the thread.

Exit System Call:

ThreadFinish is implemented by finishing the current Thread.

Task-2:

Multi-programming:

Multi-programming is implemented by created a array of 20 char* for storing a thread of max 20 thread can be allowed to execution. All the char* can run if they are not NULL. A count variable is maintained to stored in kernel to keep track of current location till where memory is filled and to load next program Address space from that location.

Round-Robin Scheduling:

When Time quantum expires, corresponding thread is yielded, and scheduler pick the next thread to execute. The quantum is set to 400.

Task-3:

Virtual Memory with Swap-File using Inverted Page-table:

For running program even when main memory doesn't have enough space to load the complete address space into the memory. Multiprogram can be implemented by storing the a back-up in swap-file in hard disk and loading a part of program address space into main memory. If a thread requests a address a particular address we check its vpn and thread id from the ipt and check whether it is in main memory or not. If it is main memory, we translate the address the address and return the ppn back to CPU. If the vpn, thread id is not available in main memory, then page fault exception is raised. This exception is handled by checked if the main memory contains any vacant position in main memory using bitmap, then that vpn is loaded in main memory in that vacant position. Else page replacement algorithm is used to remove a current page in random and replace that page with new page.

Implementation:

Task-1:

Read System Call:

case SC_Read:

```
{  
  
    int reg4=(int)kernel->machine->ReadRegister(4);  
    int reg5=(int)kernel->machine->ReadRegister(5);  
    char* c=" reading";  
    std::cout<<"          Console Read from "<<kernel->currentThread-  
>id<<endl;  
  
    int i=0;  
    while(i<reg5)  
    {  
        if(kernel->machine->WriteMem(reg4,1,(int)(*c)))  
        {  
            reg4=reg4+1;  
        }  
        i++;  
        c++;  
    }  
}
```

```

        {

/* set previous programm counter (debugging only)*/
kernel->machine->WriteRegister(PrevPCReg, kernel->machine->ReadRegister(PCReg));

/* set programm counter to next instruction (all Instructions are 4 byte wide)*/
kernel->machine->WriteRegister(PCReg, kernel->machine->ReadRegister(PCReg) + 4);

/* set next programm counter for brach execution */
kernel->machine->WriteRegister(NextPCReg, kernel->machine->ReadRegister(PCReg)+4);
        }

        break;

```

Write System Call:

case SC_Write:

```

        {

std::cout<<"                Console Write System Call from "<<kernel->currentThread->id<<endl;

        int reg4=(int)kernel->machine->ReadRegister(4);
        int reg5=(int)kernel->machine->ReadRegister(5);
        int i=0;
        while(i<reg5)
        {

                int c;

                if(kernel->machine->ReadMem(reg4, 1, &c))
                {

                        printf("%c",c);

                }

                i++;
        }

```

```

        reg4=reg4+1;
    }

    std::cout<<endl;

    {
        /* set previous programm counter (debugging only)*/
        kernel->machine->WriteRegister(PrevPCReg, kernel->machine->ReadRegister(PCReg));

        /* set programm counter to next instruction (all Instructions are 4 byte wide)*/
        kernel->machine->WriteRegister(PCReg, kernel->machine->ReadRegister(PCReg) + 4);

        /* set next programm counter for brach execution */
        kernel->machine->WriteRegister(NextPCReg, kernel->machine->ReadRegister(PCReg)+4);
    }

    break;
}

```

Fork System Call:

```

void
Forked(int num)
{
    std::cout<<"_____Forked Thread
execution_____ "<<endl;
    // ThreadExit(0);
}

case SC_ThreadFork:
{
    // std::cout<<"inside ThreadFork"<<endl;
    int reg4=kernel->machine->ReadRegister(4);
    Thread* t = new Thread("Fork ");
    t->Fork((VoidFunctionPtr)Forked,0);
    kernel->machine->WriteRegister(PrevPCReg, kernel->machine->ReadRegister(PCReg));
}

```

```

        kernel->machine->WriteRegister(PCReg, kernel->machine->ReadRegister(PCReg)
+ 4);
        kernel->machine->WriteRegister(NextPCReg, kernel->machine-
>ReadRegister(PCReg)+4);
        //std::cout<<"\n"<<kernel->currentThread->getName()<<endl;
        // kernel->scheduler->Print();
        //kernel->currentThread->StackAllocate((VoidFunctionPtr)reg4,0);

        //kernel->scheduler->Print();
        break;
    }

```

Yield System Call:

case SC_ThreadYield:

```

    {
        std::cout<<"                Thread yeild:: "<< kernel->currentThread-
>id<<endl;

        kernel->machine->WriteRegister(PrevPCReg, kernel->machine-
>ReadRegister(PCReg));
        kernel->machine->WriteRegister(PCReg, kernel->machine->ReadRegister(PCReg)
+ 4);
        kernel->machine->WriteRegister(NextPCReg, kernel->machine-
>ReadRegister(PCReg)+4);
        kernel->currentThread->Yield();

        break;
    }

```

Exit System Call:

case SC_ThreadExit:

```

    {
        std::cout<<"-----
"<<endl;
        std::cout<<"exiting the current thread : "<< kernel->currentThread->id<<endl;
        std::cout<<"-----
"<<endl;
        //kernel->scheduler->Print();
        kernel->currentThread->Finish();
        // kernel->machine->track->Print();
    }

```

```
        break;
    }
}
```

Task-2:

Multi-programming:

```
char *userProgName[20];
for(p=0; p<20;p++)
{
    userProgName[p] = NULL;
}
if (strcmp(argv[i], "-x") == 0) {
    ASSERT(i + 1 < argc);
    // char* tem=argv[i + 1];
    // userProgName.push_back(tem);
    userProgName[p]=argv[i + 1];
    p++;
    // delete tem;
    i++;
}
for(int p=0;p<20;p++)
{
    if (userProgName[p] != NULL) {
        Thread *t1 = new Thread(userProgName[p]);
        t1->Fork((VoidFunctionPtr) RunUserProg, (void *) userProgName[p]);
    }
}

pageTable = new TranslationEntry[NumPhysPages];
for (int i = 0; i < NumPhysPages; i++) {
    pageTable[i].virtualPage = i; // for now, virt page # = phys page #
    pageTable[i].physicalPage = kernel->position+i;
    pageTable[i].valid = TRUE;
    pageTable[i].use = FALSE;
    pageTable[i].dirty = FALSE;
    pageTable[i].readOnly = FALSE;
}

// zero out the entire address space
bzero(&kernel->machine->mainMemory[(kernel->position)*PageSize], size);

DEBUG(dbgAddr, "Initializing address space: " << numPages << ", " << size);
```



```

// then, copy in the code and data segments into memory
// Note: this code assumes that virtual address = physical address
if (noffH.code.size > 0) {
    DEBUG(dbgAddr, "Initializing code segment.");
    DEBUG(dbgAddr, noffH.code.virtualAddr << " ", " << noffH.code.size);
    executable->ReadAt(
        &(kernel->machine->mainMemory[noffH.code.virtualAddr+((kernel-
>position)*PageSize))),
        noffH.code.size, noffH.code.inFileAddr);
}
if (noffH.initData.size > 0) {
    DEBUG(dbgAddr, "Initializing data segment.");
    DEBUG(dbgAddr, noffH.initData.virtualAddr << " ", " << noffH.initData.size);
    executable->ReadAt(
        &(kernel->machine->mainMemory[noffH.initData.virtualAddr+((kernel-
>position)*PageSize))),
        noffH.initData.size, noffH.initData.inFileAddr);
}

#ifdef RDATA
if (noffH.readonlyData.size > 0) {
    DEBUG(dbgAddr, "Initializing read only data segment.");
    DEBUG(dbgAddr, noffH.readonlyData.virtualAddr << " ", " << noffH.readonlyData.size);
    executable->ReadAt(
        &(kernel->machine->mainMemory[noffH.readonlyData.virtualAddr+((kernel-
>position)*PageSize))),
        noffH.readonlyData.size, noffH.readonlyData.inFileAddr);
}

```

Round-Robin Scheduling:

```

void
Timer::SetInterrupt()
{
    if (!disable) {
        int delay = TimerTicks;

```

```

    if (randomize) {
        delay = 1 + (RandomNumber() % (TimerTicks * 2));
    }
    // schedule the next timer device interrupt
    kernel->interrupt->Schedule(this, 400, TimerInt);
}
}

void
Interrupt::changeInterrupt(int t)
{
    ListIterator<PendingInterrupt *> *iter = new ListIterator<PendingInterrupt *>(pending);
    PendingInterrupt* p;
    while( !iter->IsDone()) {
        PendingInterrupt* temp= iter->Item();
        iter->Next();
        if(temp->type==TimerInt)
        {
            pending->Remove(temp);
            p=temp;
        }

        //std::cout<<"Todo Elements: "<<kernel->todo->NumInList()<<endl;
    }
    p->when=kernel->stats->totalTicks+t;
    pending->Insert(p);
    delete iter;
}

```

```
}  
kernel->interrupt->changeInterrupt(400);in scheduler->Run()
```

Task-3:

```
void  
AddrSpace::Clear(int id)  
{  
    // std::cout<<"Before bitmap of thread : "<<id<<endl;  
    // kernel->machine->track->Print();  
  
    for(int i=0;i<NumPhysPages;i++)  
    {  
        IPTEntry* t = kernel->machine->ipt.at(i);  
        if(t->Thread_id==id)  
        {  
            kernel->machine->track->Clear(i);  
            t->Thread_id=-1;  
            t->virtualPage=-1;  
            t->physicalPage=-1;  
        }  
    }  
    // std::cout<<"clearing bitmap of thread : "<<id<<endl;  
    // kernel->machine->track->Print();  
}  
  
bool  
AddrSpace::Load(char *fileName)  
{  
    OpenFile *executable = kernel->fileSystem->Open(fileName);
```

```

NoffHeader noffH;

unsigned int size;

if (executable == NULL) {
    cerr << "Unable to open file " << fileName << "\n";
    return FALSE;
}

executable->ReadAt((char *)&noffH, sizeof(noffH), 0);

if ((noffH.noffMagic != NOFFMAGIC) &&
    (WordToHost(noffH.noffMagic) == NOFFMAGIC))
    SwapHeader(&noffH);

ASSERT(noffH.noffMagic == NOFFMAGIC);

#ifdef RDATA
// how big is address space?

size = noffH.code.size + noffH.readonlyData.size + noffH.initData.size +
    noffH.uninitData.size + UserStackSize;

// std::cout<<noffH.code.size <<" " <<noffH.readonlyData.size <<" " <<noffH.initData.size <<" " <<
noffH.uninitData.size <<" " <<UserStackSize<<endl;

// we need to increase the size

// to leave room for the stack

#else
// how big is address space?

size = noffH.code.size + noffH.initData.size + noffH.uninitData.size
    + UserStackSize; // we need to increase the size

// to leave room for the stack

```

```

    // std::cout<< noffH.code.size << "    "<< noffH.initData.size << " "<< noffH.uninitData.size << "
"<<UserStackSize<<endl;

#endif

    numPages = divRoundUp(size, PageSize);

    size = numPages * PageSize;

    // std::cout<<kernel->currentThread->getName()<<" Num of Pages :"<< numPages<<" , Total
:"<<size<<endl;

    ASSERT((kernel->position)+numPages <= NumPhysPages);          // check we're not trying
                                                                    // to run anything too big --
                                                                    // at least until we have
                                                                    // virtual memory

    pageTable = new TranslationEntry[NumPhysPages];
    for (int i = 0; i < NumPhysPages; i++) {
        pageTable[i].virtualPage = i;    // for now, virt page # = phys page #
        pageTable[i].physicalPage = kernel->position+i;
        pageTable[i].valid = TRUE;
        pageTable[i].use = FALSE;
        pageTable[i].dirty = FALSE;
        pageTable[i].readOnly = FALSE;
    }

    // zero out the entire address space
    // bzero(&kernel->machine->mainMemory[(kernel->position)*PageSize], size);

    DEBUG(dbgAddr, "Initializing address space: " << numPages << " , " << size);

    // then, copy in the code and data segments into memory
    // Note: this code assumes that virtual address = physical address

```

```

// std::cout<<"before add in code "<<kernel->swapFile->Length() <<"code size
:"<<noffH.code.size+UserStackSize;

kernel->loc[kernel->currentThread->id]=kernel->swapFile->Length();

int page_count=0;

if (noffH.code.size > 0) {

    DEBUG(dbgAddr, "Initializing code segment.");

    DEBUG(dbgAddr, noffH.code.virtualAddr << ", " << noffH.code.size);

    // executable->ReadAt(&(kernel->machine->mainMemory[noffH.code.virtualAddr+((kernel-
>position)*PageSize])), noffH.code.size, noffH.code.inFileAddr);

    int len=divRoundUp(noffH.code.size+UserStackSize, PageSize);

    for(int i=0;i<len;i++)

    {

        char* cod = new char[PageSize];

        executable->ReadAt(&cod[0], PageSize, noffH.code.inFileAddr+(PageSize*i));

        kernel->swapFile->WriteAt(cod, PageSize, kernel->swap_pos);

        int ppage=kernel->machine->track->FindAndSet();

        if(ppage!=-1)

        {

            IPTEntry* t= kernel->machine->ipt.at(ppage);

            t->virtualPage=pg;

            t->physicalPage=ppage;

            t->valid=true;

            t->use=false;

            t->dirty= false;

            t->readOnly=false;

            t->Thread_id=kernel->currentThread->id;

            int b=0;

            // kernel->machine->fifo->push(ppage);

            for(int j=ppage*PageSize;j<((ppage*PageSize)+PageSize);j++)

```

```

    {
        kernel->machine->mainMemory[j]=cod[b];

        b++;
    }

    pg++;
}

kernel->swap_pos=kernel->swap_pos+PageSize;

page_count++;

delete cod;
}
}

if(noffH.uninitData.size>0)
{
    int len=divRoundUp(noffH.uninitData.size, PageSize);
    for(int i=0;i<len;i++)
    {
        char* cod = new char[PageSize];

        executable->ReadAt(&cod[0], PageSize,noffH.uninitData.inFileAddr+(PageSize*i));

        kernel->swapFile->WriteAt(cod, PageSize, kernel->swap_pos);

        int ppage=kernel->machine->track->FindAndSet();

        if(ppage!=-1)
        {
            IPTEntry* t= kernel->machine->ipt.at(ppage);

            t->virtualPage=pg;

            t->physicalPage=ppage;

            t->valid=true;

            t->use=false;

            t->dirty= false;

            t->readOnly=false;

```

```

        t->Thread_id=kernel->currentThread->id;

        int b=0;

        // kernel->machine->fifo->push(ppage);

        for(int j=ppage*PageSize;j<((ppage*PageSize)+PageSize);j++)
        {
            kernel->machine->mainMemory[j]=cod[b];

            b++;
        }

        pg++;
    }

    kernel->swap_pos=kernel->swap_pos+PageSize;

    page_count++;

    delete cod;
}

}

// std::cout<<"After add code "<<kernel->swapFile->Length() <<endl;

// std::cout<<"Executable file size  after code adding "<<kernel->swapFile->Length()<<endl;


// std::cout<<"before add inIT data "<<kernel->swapFile->Length() <<"init size :"<<noffH.initData.size;
if (noffH.initData.size > 0) {

    DEBUG(dbgAddr, "Initializing data segment.");

    DEBUG(dbgAddr, noffH.initData.virtualAddr << ", " << noffH.initData.size);

    //executable->ReadAt(&(kernel->machine->mainMemory[noffH.initData.virtualAddr+((kernel-
>position)*PageSize])),noffH.initData.size, noffH.initData.inFileAddr);

    int len=divRoundUp(noffH.initData.size, PageSize);

    for(int i=0;i<len;i++)
    {

        char* dat = new char[PageSize];

        executable->ReadAt(&dat[0], PageSize, noffH.initData.inFileAddr+(PageSize*i));
    }
}

```



```

kernel->swapFile->WriteAt(dat, noffH.initData.size, kernel->swap_pos);

int ppage=kernel->machine->track->FindAndSet();
if(ppage!=-1)
{
    int b=0;

    IPTEntry* t= kernel->machine->ipt.at(ppage);

    t->virtualPage=pg;
    t->physicalPage=ppage;
    t->valid=true;
    t->use=false;
    t->dirty= false;
    t->readOnly=false;
    t->Thread_id=kernel->currentThread->id;
    // kernel->machine->fifo->push(ppage);
    for(int j=ppage*PageSize;j<((ppage*PageSize)+PageSize);j++)
    {
        kernel->machine->mainMemory[j]=dat[b];

        b++;
    }
}

kernel->swap_pos=kernel->swap_pos+PageSize;

delete dat;
}
}

// std::cout<<"After Init data "<<kernel->swapFile->Length() <<endl;

// std::cout<<"Executable file size  after data  adding "<<kernel->swapFile->Length()<<endl;

// std::cout<<"before add Read only  data "<<kernel->swapFile->Length() <<" rad only size
:"<<noffH.readonlyData.size;

#ifdef RDATA

```

```

if (noffH.readonlyData.size > 0) {
    DEBUG(dbgAddr, "Initializing read only data segment.");

    DEBUG(dbgAddr, noffH.readonlyData.virtualAddr << " , " << noffH.readonlyData.size);

    // executable->ReadAt(&(kernel->machine-
>mainMemory[noffH.readonlyData.virtualAddr+((kernel->position)*PageSize)],noffH.readonlyData.size,
noffH.readonlyData.inFileAddr);

    int len=divRoundUp(noffH.readonlyData.size, PageSize);

    for(int i=0;i<len;i++)
    {
        char* rd = new char[noffH.readonlyData.size];

        executable->ReadAt(&rd[0], noffH.readonlyData.size,
noffH.readonlyData.inFileAddr+(PageSize*i));

        kernel->swapFile->WriteAt(rd, noffH.readonlyData.size, kernel->swap_pos);

        int ppage=kernel->machine->track->FindAndSet();

        if(ppage!=-1)
        {
            int b=0;

            IPTEntry* t= kernel->machine->ipt.at(ppage);

            t->virtualPage=pg;

            t->physicalPage=ppage;

            t->valid=true;

            t->use=false;

            t->dirty= false;

            t->readOnly=false;

            t->Thread_id=kernel->currentThread->id;

            // kernel->machine->fifo->push(ppage);

            for(int j=ppage*PageSize;j<((ppage*PageSize)+PageSize);j++)
            {
                kernel->machine->mainMemory[j]=rd[b];

                b++;
            }
        }
    }
}

```

```

        }
    }
    kernel->swap_pos=kernel->swap_pos+PageSize;
    delete rd;
}

}
#endif

// std::cout<<"After Read data "<<kernel->swapFile->Length() <<endl;
std::cout<<"Swap Space starting at "<<kernel->loc[kernel->currentThread->id]<<" ending at
"<<kernel->swapFile->Length()<<"\n"<<endl;

// kernel->machine->track->Print();

// kernel->position= kernel->position+numPages;

for(int i=0;i<NumPhysPages;i++)
{
    IPTEntry* t= kernel->machine->ipt.at(i);

    // std::cout<<t->virtualPage<<" "<<t->physicalPage<<"Thread id"<<t->Thread_id<<" readOnly "<<t-
>readOnly<<endl;

}

delete executable;                // close file
return TRUE;                      // success
}

```

ExceptionType

Machine::Translate(int virtAddr, int* physAddr, int size, bool writing)

```

{
    // std::cout<<kernel->currentThread->id<<"   inside Machine translate   "<<virtAddr<<endl;

    int i;

    unsigned int vpn,offset;

    unsigned int ppn;

    ppn=190;

    IPTEntry* entry=NULL;


    DEBUG(dbgAddr, "\tTranslate " << virtAddr << (writing ? " , write" : " , read"));
    if (((size == 4) && (virtAddr & 0x3)) || ((size == 2) && (virtAddr & 0x1))){
        DEBUG(dbgAddr, "Alignment problem at " << virtAddr << " , size " << size);
        return AddressErrorException;
    }


    vpn = (unsigned) virtAddr / PageSize;
    offset = (unsigned) virtAddr % PageSize;


    // vpn=divRoundUp(virtAddr, PageSize)-1;
    // offset=virtAddr-vpn*PageSize;


    if (tlb == NULL) {    // => page table => vpn is index into table
        // if (vpn >= kernel->currentThread->space->numPages)
        // {
        //     DEBUG(dbgAddr, "Illegal virtual page # " << virtAddr);
        //     return AddressErrorException;
        // }
        // else if (!pageTable[vpn].valid)
        // {
        //     DEBUG(dbgAddr, "Invalid virtual page # " << virtAddr);

```

```

// return PageFaultException;
// }
// entry = &pageTable[vpn];
bool v= false;
for(int i=0;i<NumPhysPages;i++)
{
    IPTEntry* t= ipt.at(i);
    if(t->Thread_id==kernel->currentThread->id && vpn==t->virtualPage)
    {
        entry=t;
        v= true;
        ppn=entry->physicalPage;
        break;
    }
}
if(entry==NULL)
{
    PageFaultException;
}
}

// std::cout<<"vpn    : "<< vpn<<" ;ppn : "<<ppn<<" virtAddr : "<<virtAddr<<" id "<<kernel-
>currentThread->id<<endl;

if(ppn>NumPhysPages)
{
    return PageFaultException;

}

if (entry->readOnly && writing) { // trying to write to a read-only page

```

```

    DEBUG(dbgAddr, "Write to read-only page at " << virtAddr);
    // std::cout<<"vpn  : "<< vpn<<" ;ppn  : "<<ppn<<" virtAddr  : " <<virtAddr<<endl;
    // std::cout<<entry->readOnly << writing<<endl;
    return ReadOnlyException;
}

if (ppn >= NumPhysPages) {
    DEBUG(dbgAddr, "Illegal pageframe " << ppn);
    return BusErrorException;
}

entry->use = TRUE;    // set the use, dirty bits
if (writing)
    entry->dirty = TRUE;
*physAddr = ppn * PageSize + offset;
ASSERT((*physAddr >= 0) && ((*physAddr + size) <= MemorySize));
DEBUG(dbgAddr, "phys addr = " << *physAddr);
return NoException;

}

case PageFaultException:
{
    std::cout<<"                Page Fault Exception raised from " <<kernel->currentThread-
>id<<endl;

    int out=kernel->machine->track->FindAndSet();

```

```

if(out!=-1)
{
    IPTEntry* p = kernel->machine->ipt.at(out);
    int addr=kernel->machine->ReadRegister(BadVAddrReg);
    unsigned int vpn1= (unsigned) addr / PageSize;
    char* data1= new char[PageSize];
    int nt=kernel->currentThread->id;
    int nt_pos= kernel->loc[nt];
    kernel->swapFile->ReadAt(&data1[0], PageSize,nt_pos+vpn1*PageSize);
    int b=0;
    for(int j=out*PageSize;j<((out*PageSize)+PageSize);j++)
    {
        kernel->machine->mainMemory[j]=data1[b];
        b++;
    }
    p->virtualPage=vpn1;
    p->physicalPage=out;
    p->valid=true;
    p->use=false;
    p->dirty= false;
    p->readOnly=false;
    p->Thread_id=kernel->currentThread->id;
}
else
{
    out= rand()%70+10;
    kernel->machine->track->Clear(out);
    IPTEntry* p = kernel->machine->ipt.at(out);
    int t=p->Thread_id;

```

```

    int file_pos= kernel->loc[t];

    char* data= new char[PageSize];

    int main_pos=out*PageSize;

    int vpn=p->virtualPage;

    for(int j=0;j<PageSize;j++)
    {
        // kernel->machine->mainMemory[j]=cod[b];

        data[j]=kernel->machine->mainMemory[main_pos];

        main_pos++;
    }

    kernel->swapFile->WriteAt(data, PageSize, file_pos+vpn*PageSize);

    delete data;

    // std::cout<<kernel->swapFile->Length()<<endl;

    int addr=kernel->machine->ReadRegister(BadVAddrReg);

    unsigned int vpn1= (unsigned) addr / PageSize;

    char* data1= new char[PageSize];

    int nt=kernel->currentThread->id;

    int nt_pos= kernel->loc[nt];

    kernel->swapFile->ReadAt(&data1[0], PageSize,nt_pos+vpn1*PageSize);

    int b=0;

    for(int j=out*PageSize;j<((out*PageSize)+PageSize);j++)
    {
        kernel->machine->mainMemory[j]=data1[b];

        b++;
    }

    p->virtualPage=vpn1;

    p->physicalPage=out;

    p->valid=true;

    p->use=false;

```



```

p->dirty= false;

p->readOnly=false;

p->Thread_id=kernel->currentThread->id;

kernel->machine->track->Mark(out);

    }

// kernel->machine->fifo->push(out);


    break;

}

```

Testing:

- 1) Make in coff2noff folder
- 2) Make clean, make distclean and make in test folder
- 3) Make test in test program
- 4) Make nachos in nachos folder.
- 5) Maximum of 20 programs can be allowed to run in parallel.
- 6) If creating any new user programs, create in test copying corresponding system calls from test user program and make as above.
- 7) Run `./nachos -x ../test/matmult -x ../test/test/ -x ../test/matmult -x ../test/matmult -x ../test/matmult -x ../test/matmult`
- 8) Each thread will 400 quantum time and yield and all thread will finish once execution is done with exit codes.

Output:

```

./nachos -x ../test/matmult -x ../test/matmult -x ../test/matmult -x ../test/test -x ../test/test -x
../test/add -x ../test/add

```

Current Thread ID : 1

Current Thread ID : 1

Swap Space starting at 0 ending at 7040

Current Thread ID : 2

Swap Space starting at 7040 ending at 14080

Current Thread ID : 3

Swap Space starting at 14080 ending at 21120

Page Fault Exception raised from 3

Page Fault Exception raised from 3

Page Fault Exception raised from 3

Current Thread ID : 4

Swap Space starting at 21120 ending at 22656

Page Fault Exception raised from 4

Page Fault Exception raised from 4

Page Fault Exception raised from 4

Page Fault Exception raised from 4

Console Read from 4

Console Write System Call from 4

reading

Page Fault Exception raised from 4

Thread yeild:: 4

Current Thread ID : 5

Swap Space starting at 22656 ending at 24192

Page Fault Exception raised from 5

Page Fault Exception raised from 5

Page Fault Exception raised from 5

Page Fault Exception raised from 5

Console Read from 5

Console Write System Call from 5

reading

Page Fault Exception raised from 5

Thread yeild:: 5

Current Thread ID : 6

Swap Space starting at 24192 ending at 25600

Page Fault Exception raised from 6

Page Fault Exception raised from 6

Page Fault Exception raised from 6

Page Fault Exception raised from 6

Thread yeild:: 6

Current Thread ID : 7

Swap Space starting at 25600 ending at 27008

Page Fault Exception raised from 7

Page Fault Exception raised from 7

Page Fault Exception raised from 7

Page Fault Exception raised from 7

Thread yeild:: 7

Current Thread ID : 1

Current Thread ID : 2

Page Fault Exception raised from 2

Page Fault Exception raised from 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 8

_____ Forked Thread execution _____

Current Thread ID : 4

Page Fault Exception raised from 4

exiting the current thread : 4

Current Thread ID : 9

_____ Forked Thread execution _____

Current Thread ID : 5

Page Fault Exception raised from 5

Page Fault Exception raised from 5

exiting the current thread : 5

Current Thread ID : 6

exiting the current thread : 6

Current Thread ID : 7

Page Fault Exception raised from 7

exiting the current thread : 7

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Page Fault Exception raised from 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Page Fault Exception raised from 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Page Fault Exception raised from 2

Page Fault Exception raised from 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Page Fault Exception raised from 2

Current Thread ID : 3

Page Fault Exception raised from 3

Page Fault Exception raised from 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Page Fault Exception raised from 2

Page Fault Exception raised from 2

Page Fault Exception raised from 2

Page Fault Exception raised from 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Page Fault Exception raised from 1

Page Fault Exception raised from 1

Page Fault Exception raised from 1

Current Thread ID : 2

Page Fault Exception raised from 2

Current Thread ID : 3

Page Fault Exception raised from 3

Page Fault Exception raised from 3

Current Thread ID : 1

Page Fault Exception raised from 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Page Fault Exception raised from 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Page Fault Exception raised from 2

Page Fault Exception raised from 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Page Fault Exception raised from 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Page Fault Exception raised from 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Page Fault Exception raised from 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Page Fault Exception raised from 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Page Fault Exception raised from 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Page Fault Exception raised from 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Page Fault Exception raised from 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Page Fault Exception raised from 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Page Fault Exception raised from 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Page Fault Exception raised from 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Page Fault Exception raised from 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Page Fault Exception raised from 2

Page Fault Exception raised from 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Page Fault Exception raised from 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Page Fault Exception raised from 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Page Fault Exception raised from 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Page Fault Exception raised from 3

Page Fault Exception raised from 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Page Fault Exception raised from 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Page Fault Exception raised from 2

Current Thread ID : 3

Page Fault Exception raised from 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Page Fault Exception raised from 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Page Fault Exception raised from 2

Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 1

Current Thread ID : 3

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Page Fault Exception raised from 1

Page Fault Exception raised from 1

Current Thread ID : 2

Page Fault Exception raised from 2

Current Thread ID : 3

Page Fault Exception raised from 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1

Current Thread ID : 2

Current Thread ID : 3

Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1

Exiting the current thread : 1 7220

Current Thread ID : 2

Page Fault Exception raised from 2

Exiting the current thread : 2 7220

Current Thread ID : 3

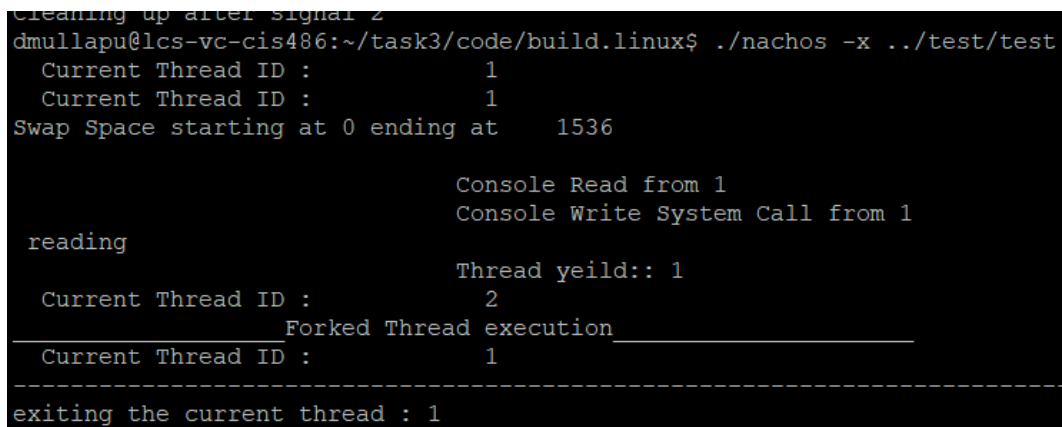
Exiting the current thread : 3 7220

^C

Cleaning up after signal 2

]0;dmullapu@lcs-vc-cis486: ~/task3/code/build.linux dmullapu@lcs-vc-
cis486:~/task3/code/build.linux\$

ScreenShots:



```
Cleaning up after signal 2
dmullapu@lcs-vc-cis486:~/task3/code/build.linux$ ./nachos -x ../test/test
Current Thread ID : 1
Current Thread ID : 1
Swap Space starting at 0 ending at 1536

Console Read from 1
Console Write System Call from 1

reading
Thread yeild:: 1
Current Thread ID : 2
Forked Thread execution
Current Thread ID : 1
-----
exiting the current thread : 1
-----
```

```
Current Thread ID :      1
Current Thread ID :      1
Swap Space starting at 0 ending at      1408
```

```
exiting the current thread : 1
```

```
Cleaning up after signal 2
```

[illegible]

```
Current Thread ID :      1  
Current Thread ID :      2  
Current Thread ID :      3  
Current Thread ID :      1  
Current Thread ID :      2  
Current Thread ID :      3  
Current Thread ID :      1  
Current Thread ID :      2  
Current Thread ID :      3  
Current Thread ID :      1  
Current Thread ID :      2  
Current Thread ID :      3  
Current Thread ID :      1  
Current Thread ID :      2  
Current Thread ID :      3  
Current Thread ID :      1  
Current Thread ID :      2  
Current Thread ID :      3  
Current Thread ID :      1  
Current Thread ID :      2  
Current Thread ID :      3  
Current Thread ID :      1  
Current Thread ID :      2  
Current Thread ID :      3  
Current Thread ID :      1  
Current Thread ID :      2  
Current Thread ID :      3  
  
Page Fault Exception raised from 1  
Current Thread ID :      2  
Current Thread ID :      3  
Current Thread ID :      1  
Current Thread ID :      2  
  
Page Fault Exception raised from 2  
Current Thread ID :      3  
Current Thread ID :      1  
Current Thread ID :      2  
Current Thread ID :      3  
Current Thread ID :      1  
Current Thread ID :      2  
Current Thread ID :      3  
Current Thread ID :      1  
Current Thread ID :      2  
Current Thread ID :      3  
Current Thread ID :      1  
Current Thread ID :      2
```

```
Page Fault Exception raised from 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Page Fault Exception raised from 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Page Fault Exception raised from 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Page Fault Exception raised from 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
Current Thread ID : 2
Current Thread ID : 3
Current Thread ID : 1
```

```
-----
Exiting the current thread : 1 7220
-----
```

```
Current Thread ID : 2
Page Fault Exception raised from 2
-----
```

```
Exiting the current thread : 2 7220
-----
```

```
Current Thread ID : 3
-----
```

```
Exiting the current thread : 3 7220
-----
```

