OPERATING SYSTEM

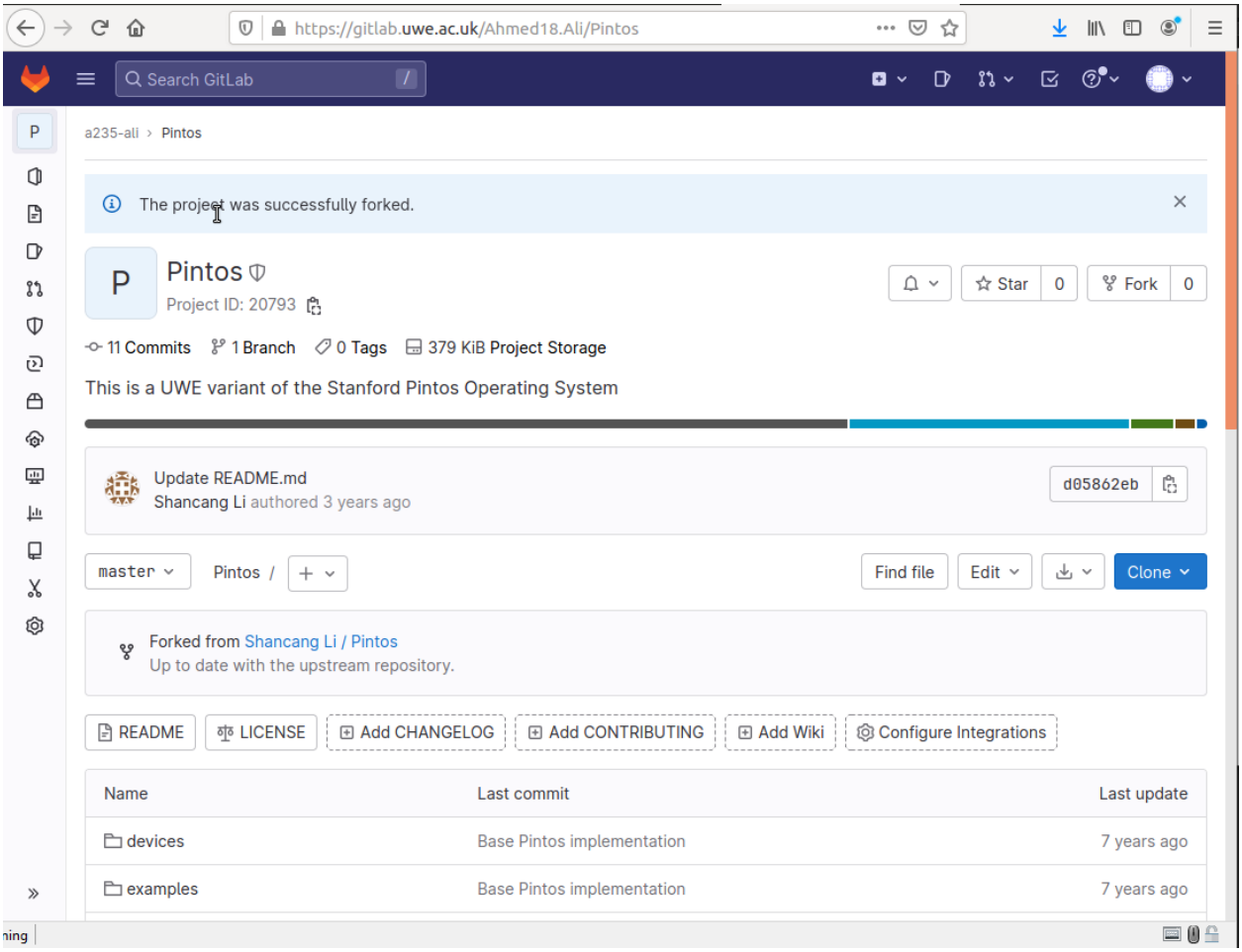
Worksheet 3 (PintOS)

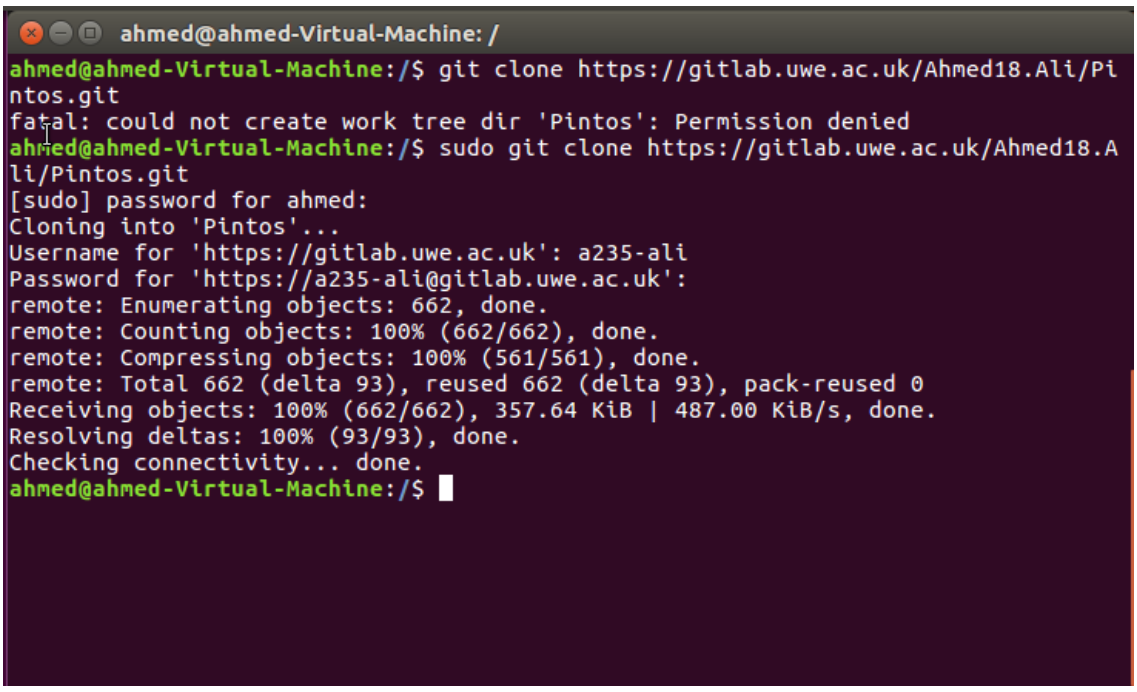
Ahmed Ali
S2201026

CONTENTS

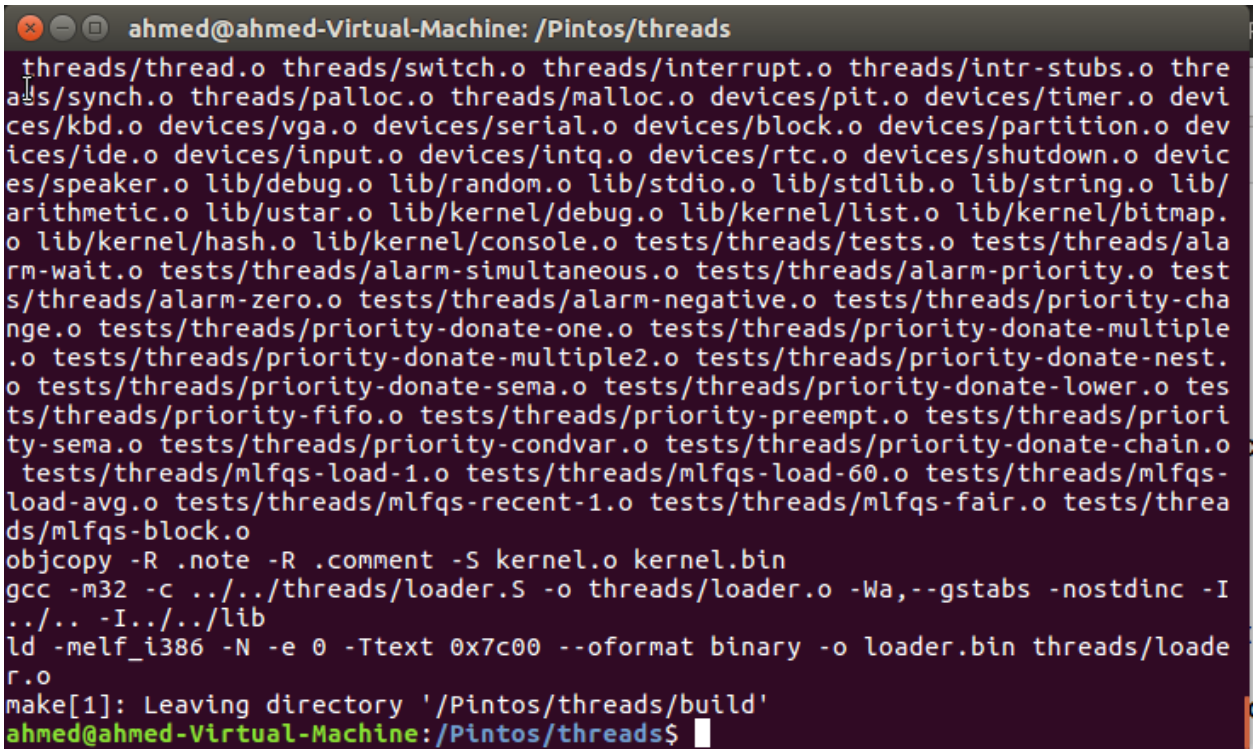
1. Download and build a local verson of Pintos OS

Fork a version of Pintos into ***your own Gitlab workspace*** and then clone the resulting repo locally.

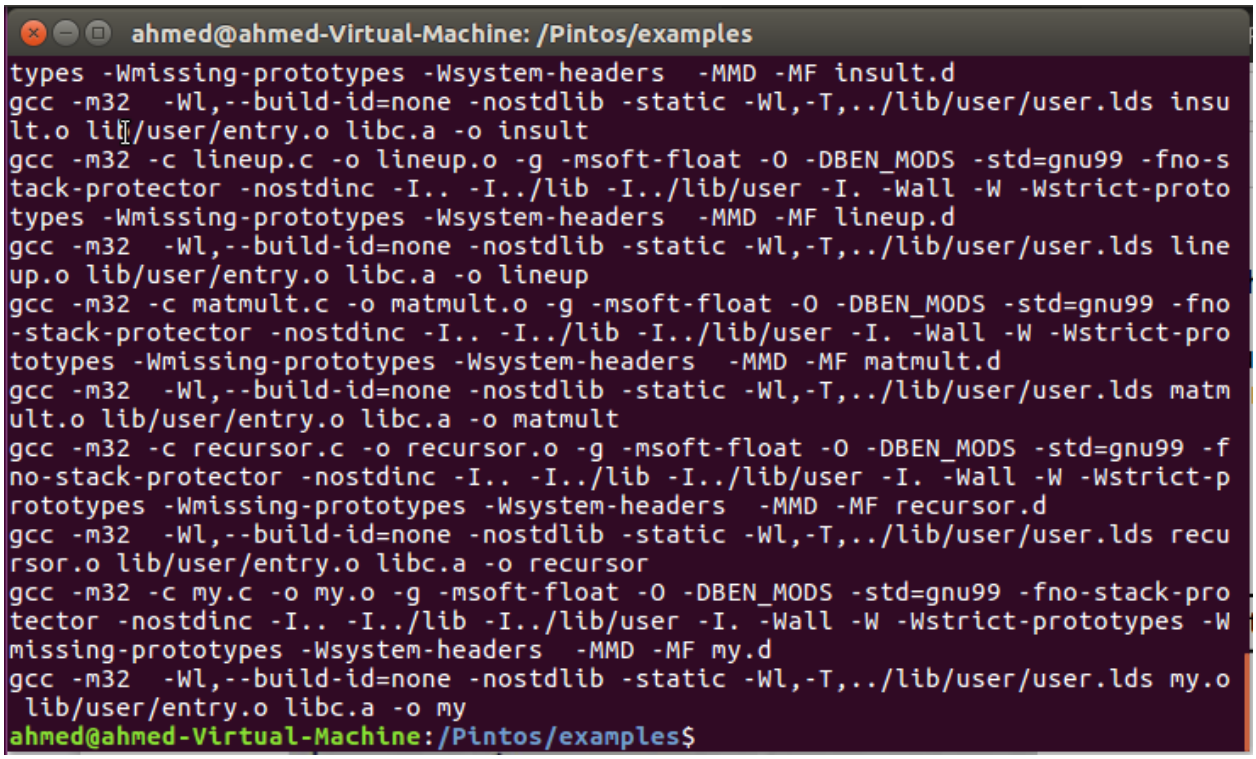




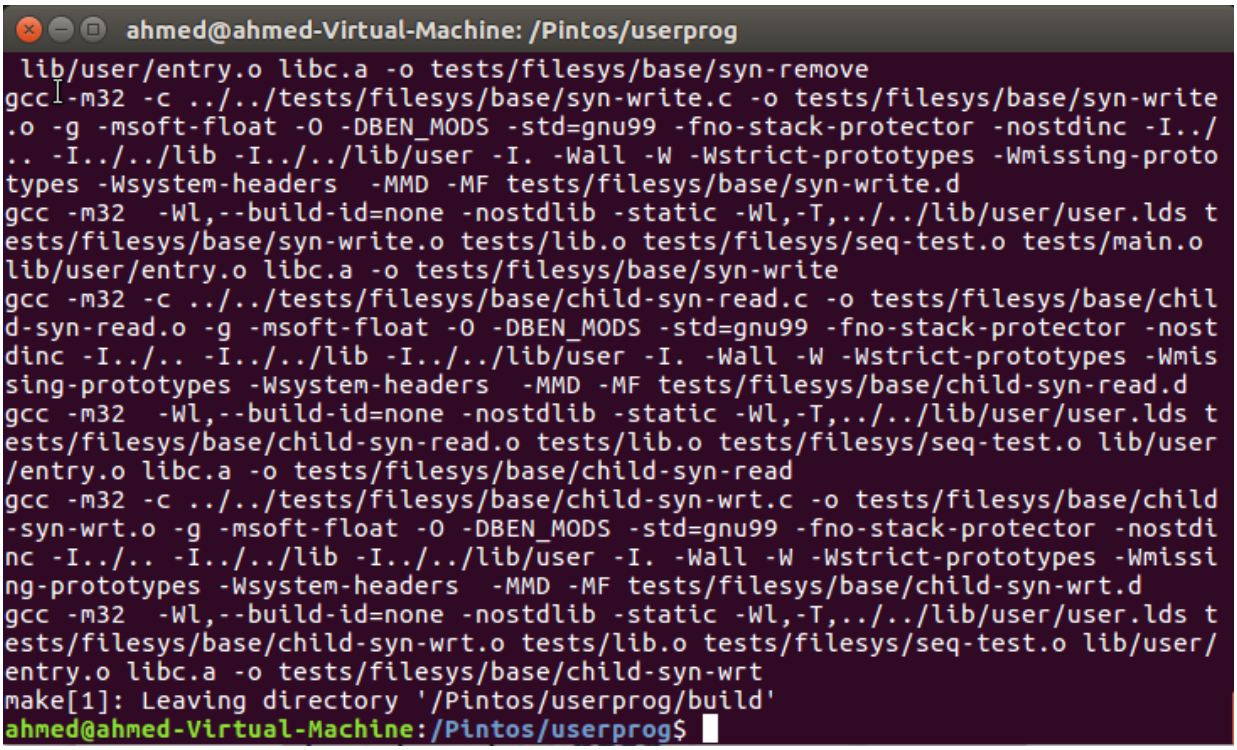
a. To build Pintos you must run make run different directories. First, cd into the **threads** directory and then issue the **make** command.



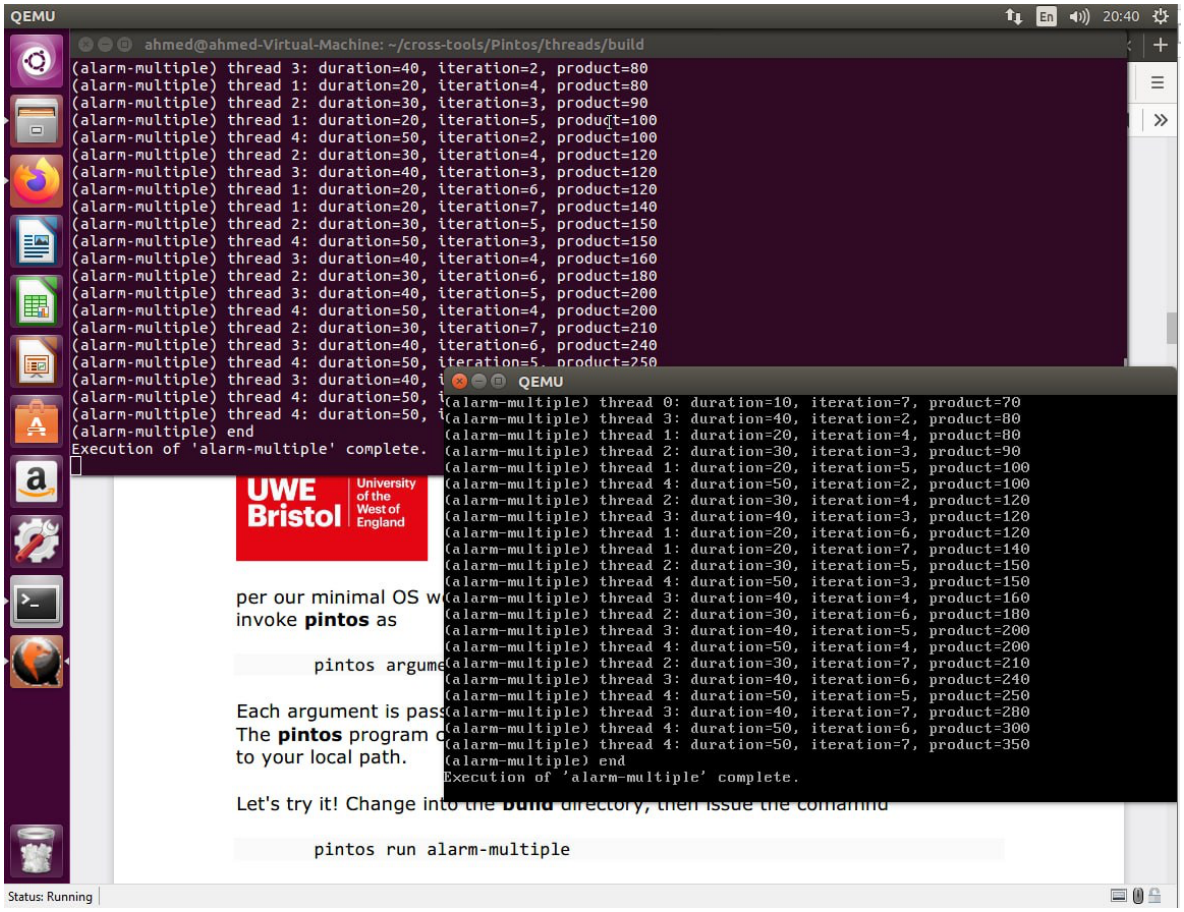
Run ‘make’ to compile examples
RESULT

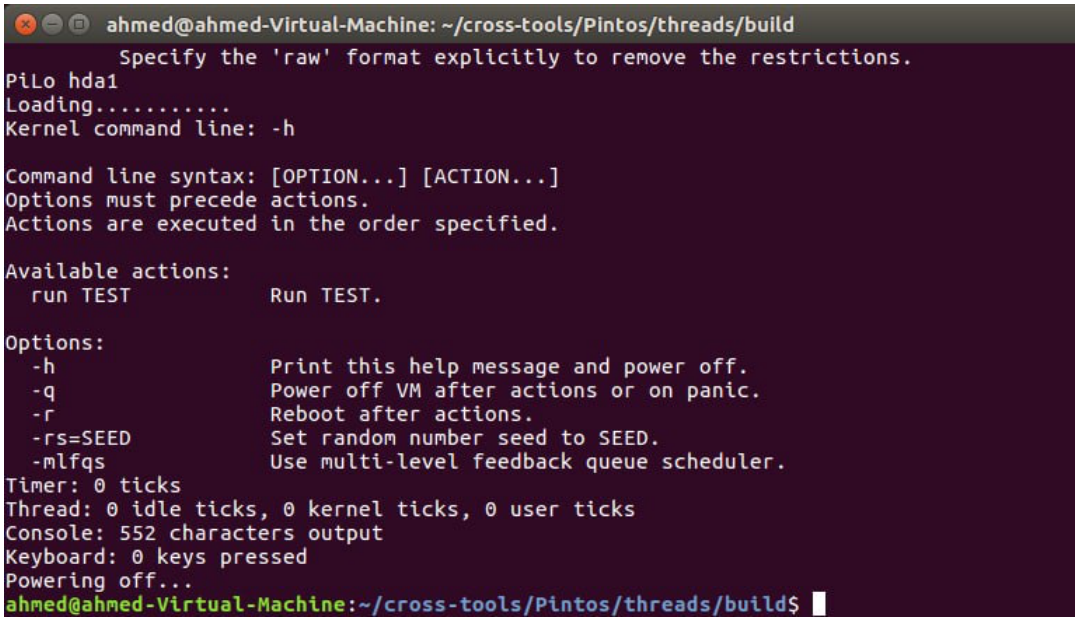


Run ‘make’ to compile user interface programs. Result as follows:



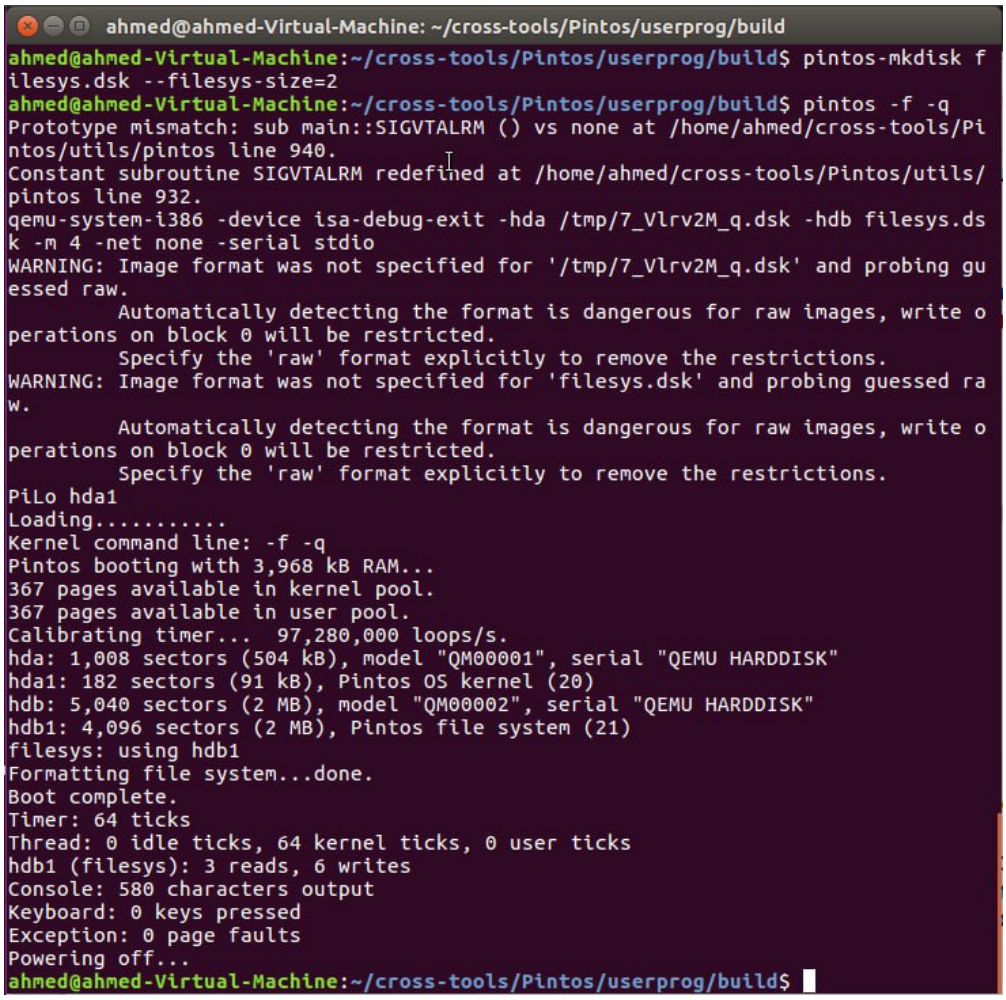
Running Pintos



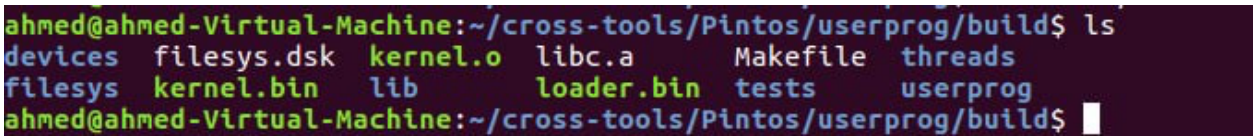


USING THE FILE SYSTEM

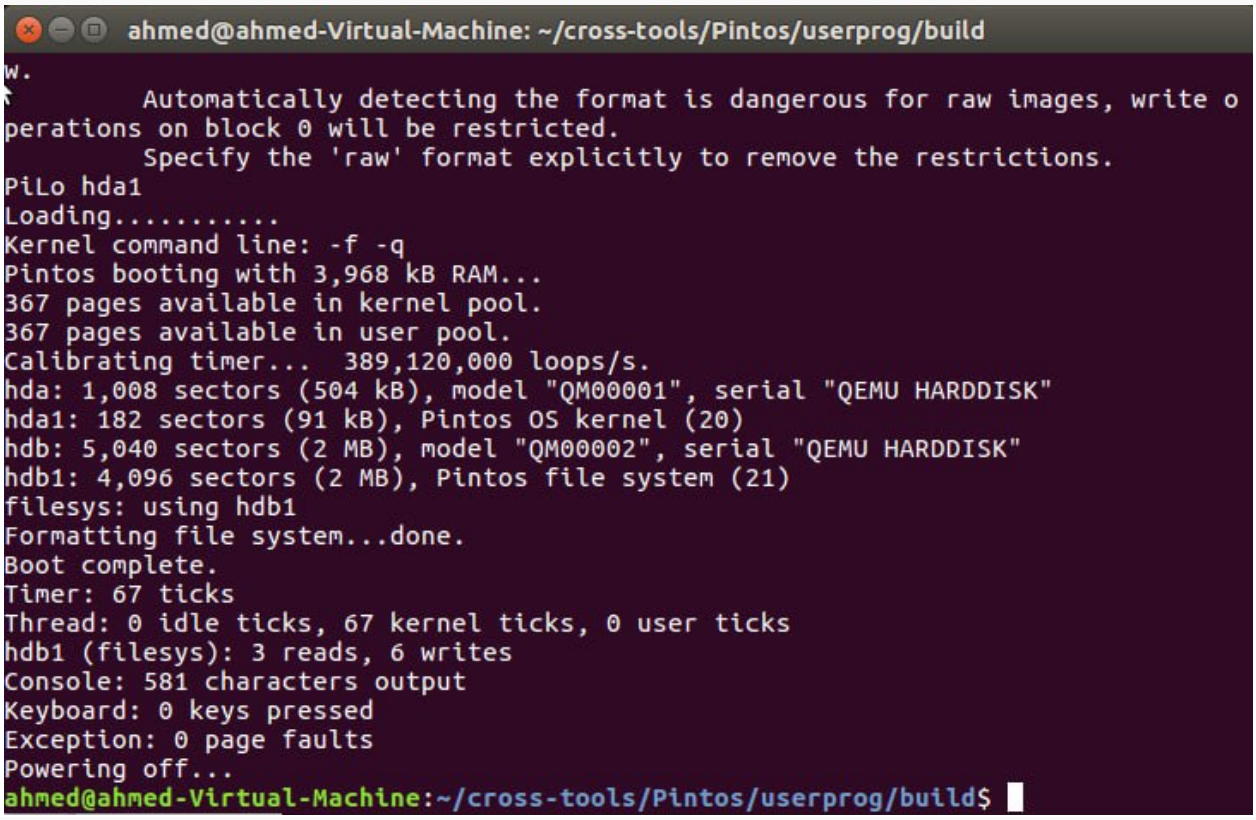
create a simulated disk with a file system partition

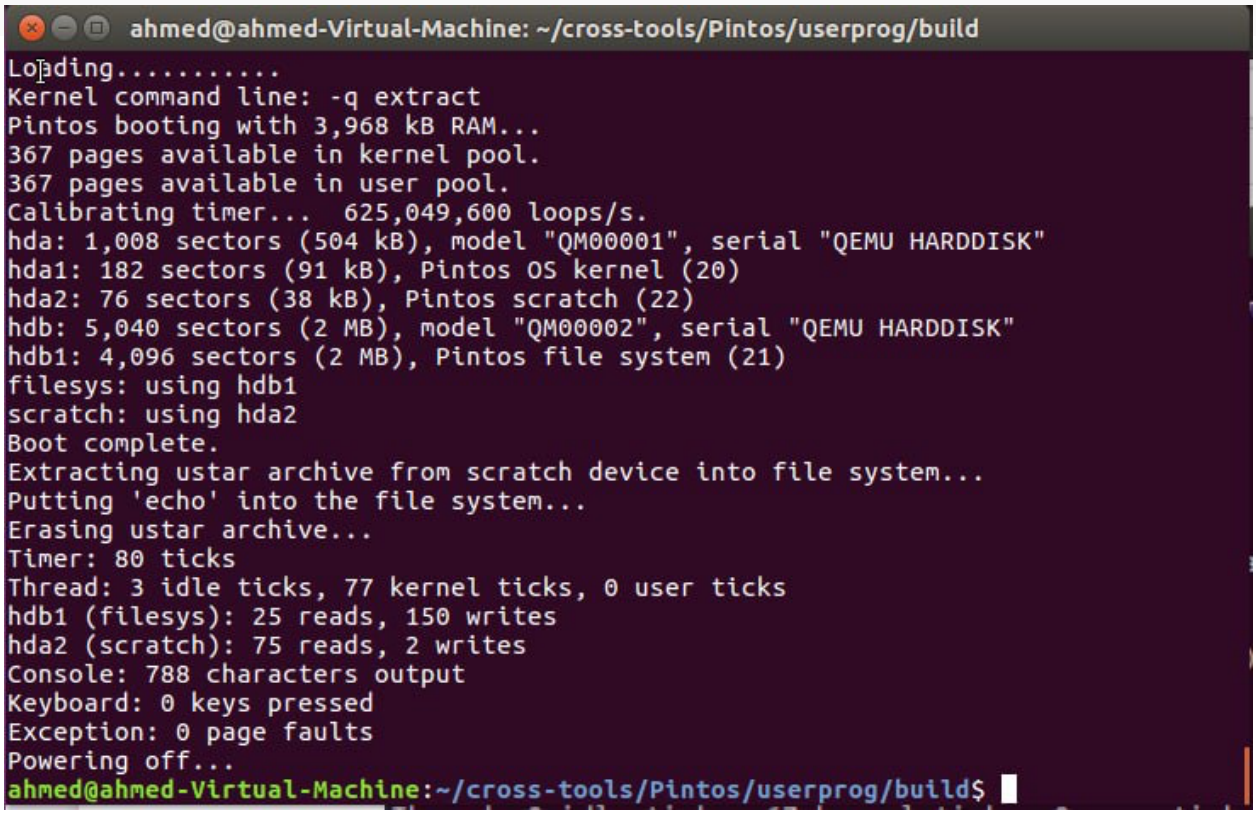


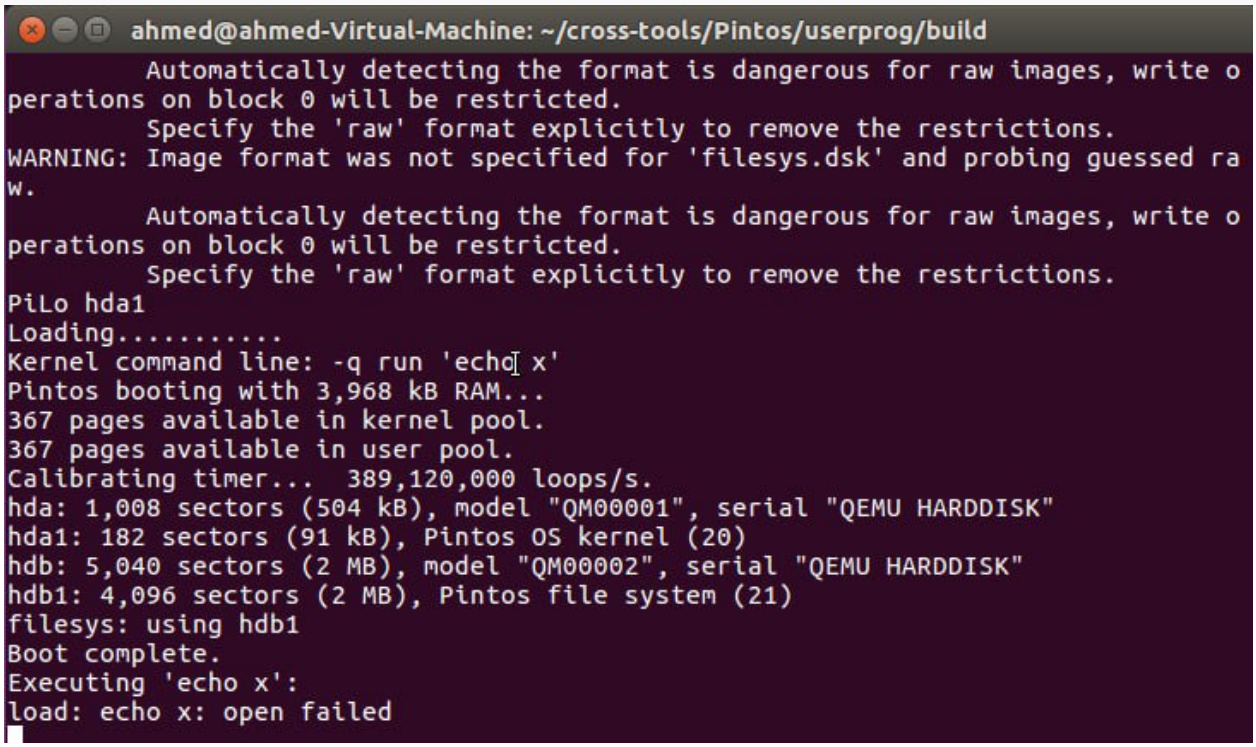
Then format the file system partition as shown above giving those results



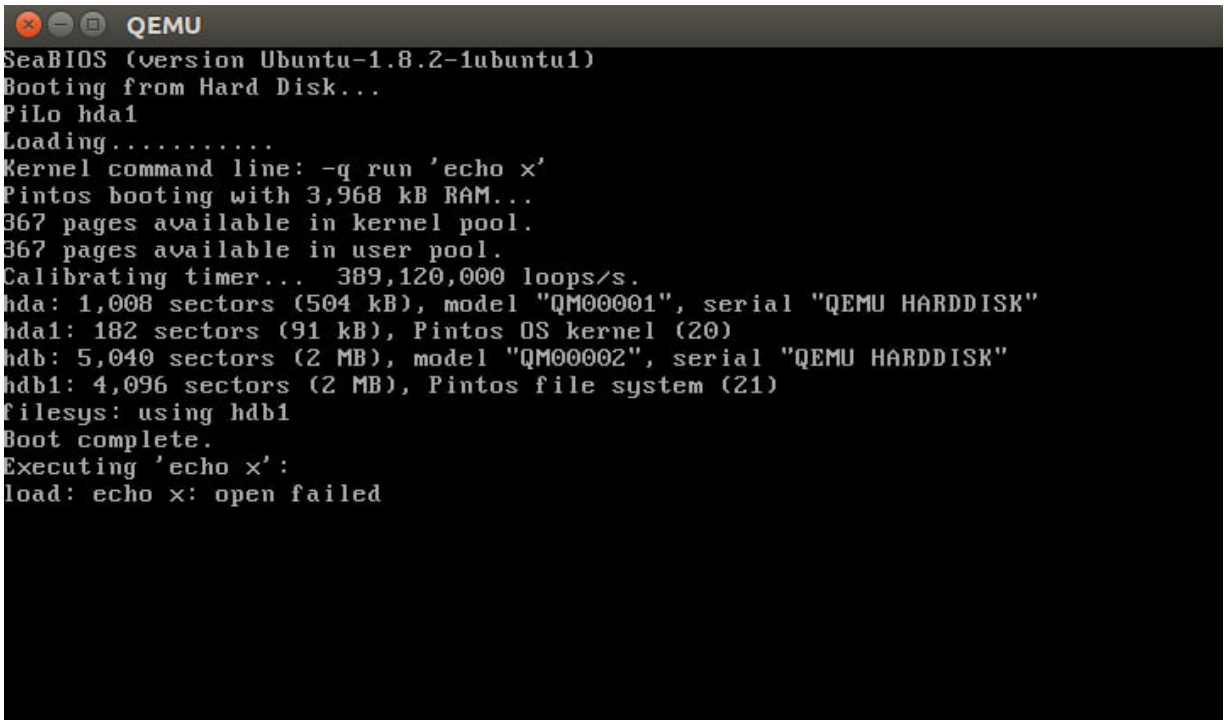
After filesys.dsk creation you need to run pintos -f -q and the following should be seen.








```
pintos -q run 'echo x'
```



EDITING CODE :

```
ahmed@ahmed-Virtual-Machine: ~/cross-tools/Pintos/userprog/build
on block 0 will be restricted.
        Specify the 'raw' format explicitly to remove the restrictions.
PiLo hda1
Loading...........
Kernel command line: -q run echo
Pintos booting with 3,968 kB RAM...
367 pages available in kernel pool.
367 pages available in user pool.
Calibrating timer...  419,020,800 loops/s.
hda: 1,008 sectors (504 kB), model "QM00001", serial "QEMU HARDDISK"
hda1: 182 sectors (91 kB), Pintos OS kernel (20)
hdb: 5,040 sectors (2 MB), model "QM00002", serial "QEMU HARDDISK"
hdb1: 4,096 sectors (2 MB), Pintos file system (21)
filesys: using hdb1
Boot complete.
Executing 'echo':
Page fault at 0xc0000008: rights violation error reading page in user context.
echo: dying due to interrupt 0x0e (#PF Page-Fault Exception).
Interrupt 0x0e (#PF Page-Fault Exception) at eip=0x80480f1
 cr2=c0000008 error=00000005
 eax=00000000 ebx=00000000 ecx=00000000 edx=00000000
 esi=00000000 edi=00000000 esp=bfffffec ebp=00000000
 cs=001b ds=0023 es=0023 ss=0023
ahmed@ahmed-Virtual-Machine:~/cross-tools/Pintos/userprog/build$
```

Now when this page fault occurs: Change *esp = PHYS_BASE;

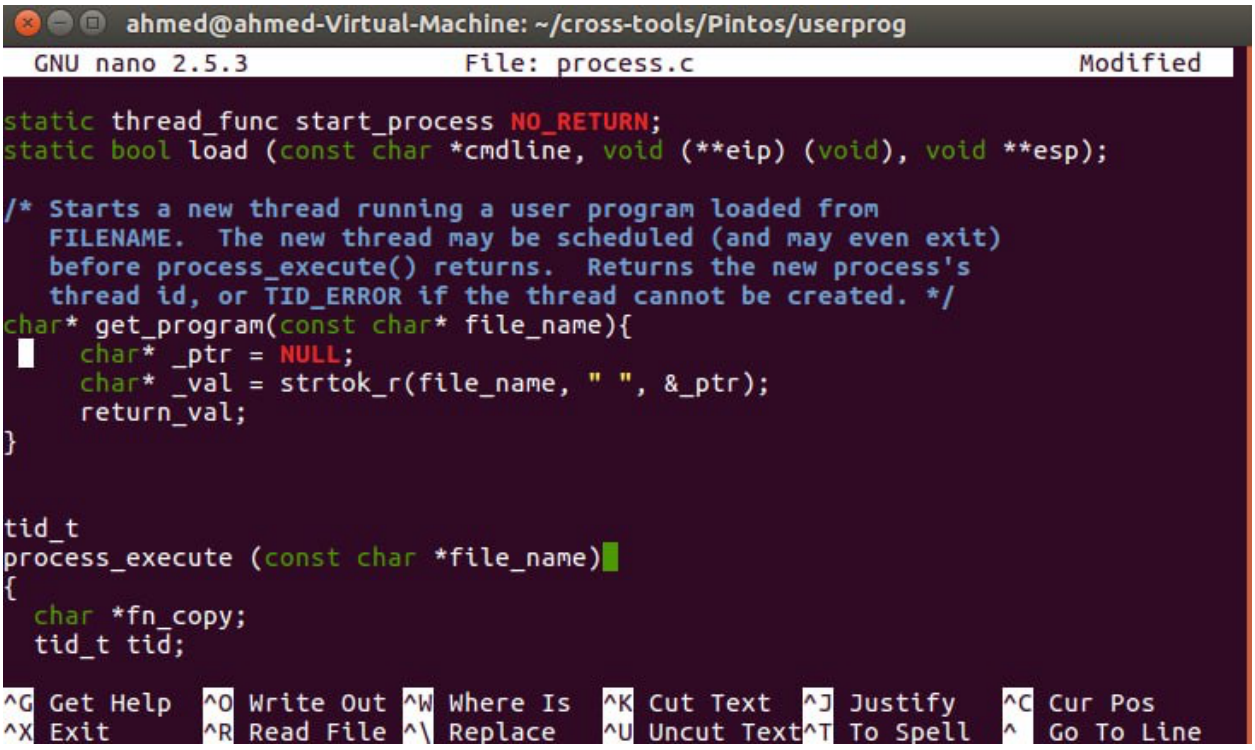
to *esp = PHYS_BASE – 12;

```
ahmed@ahmed-Virtual-Machine: ~/cross-tools/Pintos/userprog
 GNU nano 2.5.3                File: process.c                          Modified

  kpage = palloc_get_page (PAL_USER | PAL_ZERO);
  if (kpage != NULL)
    {
      success = install_page (((uint8_t *) PHYS_BASE) - PGSIZE, kpage, true);
      if (success) {
        *esp = PHYS_BASE - 12;
      } else
        palloc_free_page (kpage);
    }
  return success;
}

/* Adds a mapping from user virtual address UPAGE to kernel
   virtual address KPAGE to the page table.
   If WRITABLE is true, the user process may modify the page;
   otherwise, it is read-only.
   UPAGE must not already be mapped.
   KPAGE should probably be a page obtained from the user pool
   with palloc_get_page().

^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^_ Go To Line
```

User program .
We have to set the path to

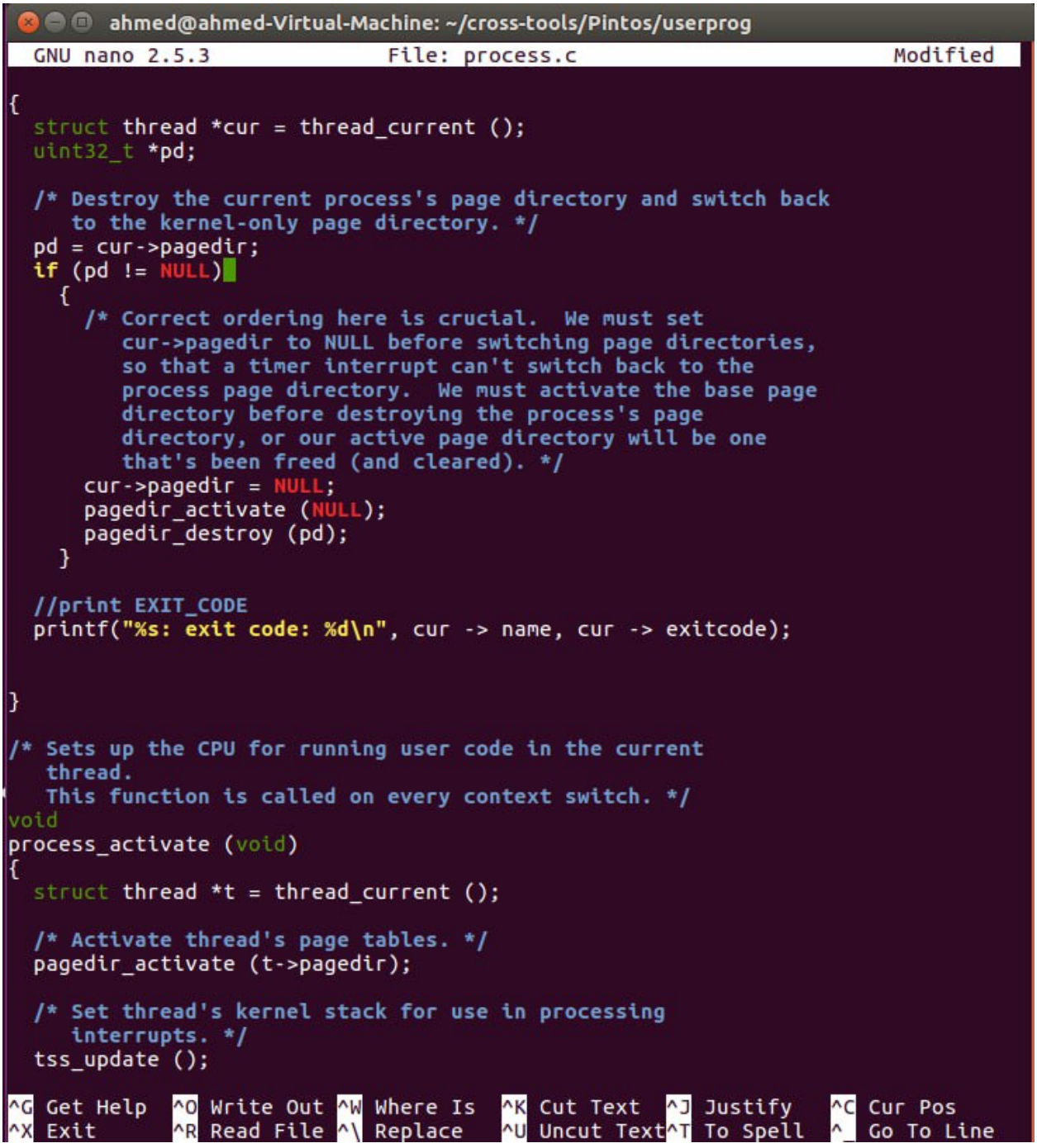
```
export PREFIX="$HOME/cross-tools"
export TARGET=i686-elf
export PATH="$PREFIX/bin:$PATH"

export PATH="$PATH:$HOME/cross-tools/Pintos/utils/"
```

Create a new function to extract the program name from an argument string that contains the program name and any number of command-line parameters.

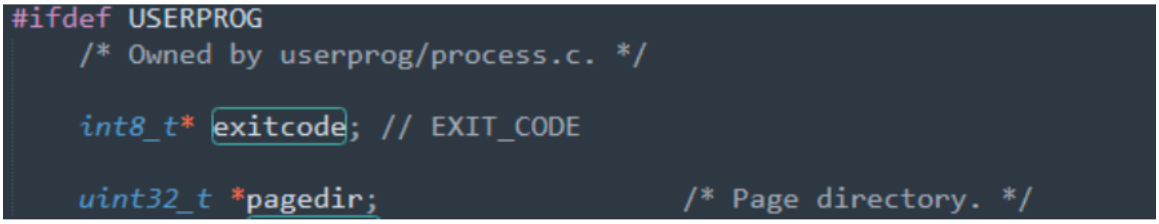


Start process to ..

To implement exit code add to process_exit ..



And to thread.h



Running – Result