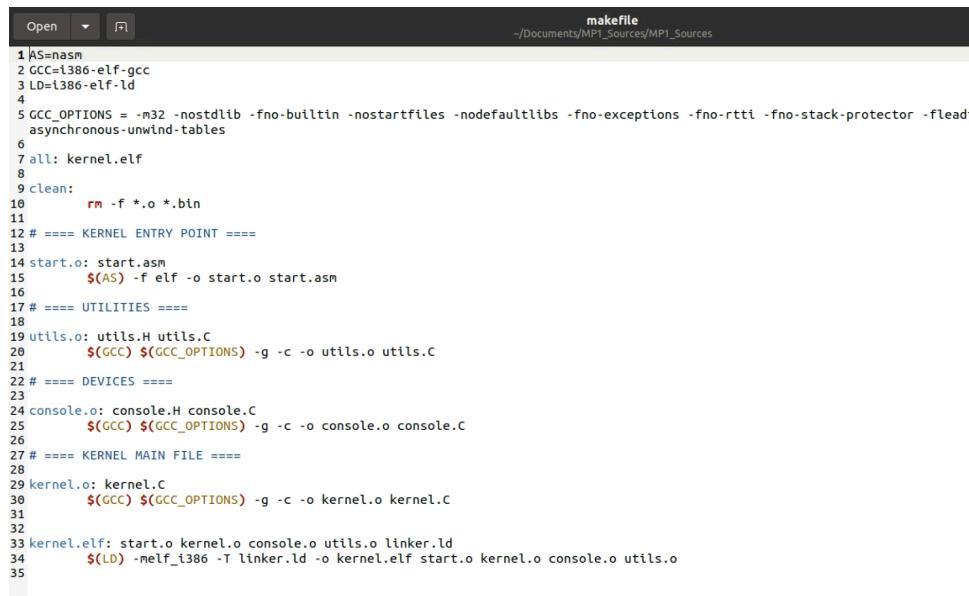


Integrating Bochs Environment with GDB: CSCE 611: MP1

To use GDB with Bochs, we need to build Bochs with gdb-stub enabled.

1. \Download Bochs with gdb sourcecode from the following location:
<https://sourceforge.net/projects/bochs/files/bochs/2.6.8/>
2. Configure Bochs with gdb stub enabled, under the directory of the Bochs source code:
 - a. sudo ./configure --enable-gdb-stub
3. Install Debian package libXrandr - provides an X Window System client interface to the RandR extension to the X protocol.
sudo apt install libxrandr-dev
4. After the configuration, to make it and move it to /usr/local/bin, run
 - a. sudo make
 - b. sudo make install
5. Then we produce an ELF (Executable and Linkable Format) output that keeps the debugging information.
6. We add the “-g” flag to our existing makefile at each of the compile steps to produce an object file containing the debug information.

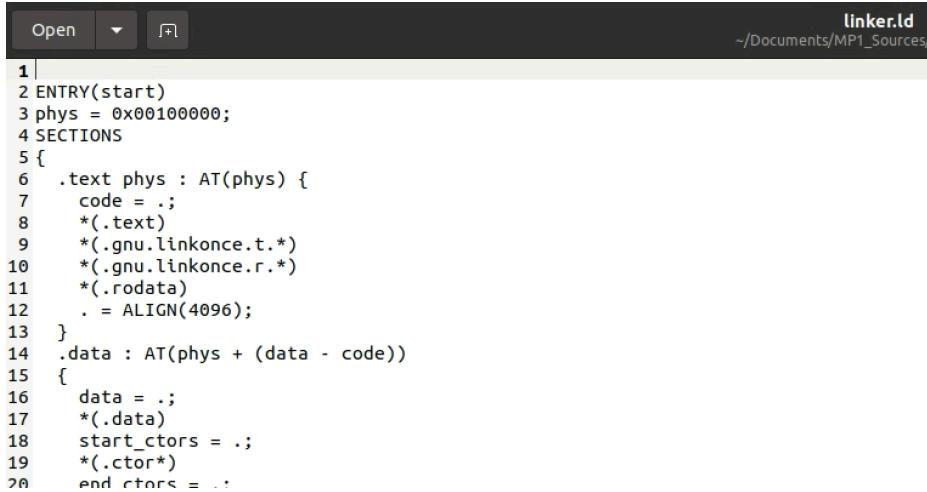


The screenshot shows a text editor window with the title bar "makefile" and the path "~/Documents/MP1_Sources/MP1_Sources". The content of the file is a makefile with the following code:

```
Open ▾ makefile
~/Documents/MP1_Sources/MP1_Sources

1 AS=nasm
2 GCC=i386-elf-gcc
3 LD=i386-elf-ld
4
5 GCC_OPTIONS = -m32 -nostdlib -fno-builtin -nostartfiles -nodefaultlibs -fno-exceptions -fno-rtti -fno-stack-protector -flead
   asynchronous-unwind-tables
6
7 all: kernel.elf
8
9 clean:
10    rm -f *.o *.bin
11
12 # ===== KERNEL ENTRY POINT =====
13
14 start.o: start.asm
15    $(AS) -f elf -o start.o start.asm
16
17 # ===== UTILITIES =====
18
19 utils.o: utils.H utils.C
20    $(GCC) $(GCC_OPTIONS) -g -c -o utils.o utils.C
21
22 # ===== DEVICES =====
23
24 console.o: console.H console.C
25    $(GCC) $(GCC_OPTIONS) -g -c -o console.o console.C
26
27 # ===== KERNEL MAIN FILE =====
28
29 kernel.o: kernel.C
30    $(GCC) $(GCC_OPTIONS) -g -c -o kernel.o kernel.C
31
32
33 kernel.elf: start.o kernel.o console.o utils.o linker.ld
34    $(LD) -melf_i386 -T linker.ld -o kernel.elf start.o kernel.o console.o utils.o
35
```

7. In the linker.ld file, we remove the first line to get the ELF output



```
linker.ld
~/Documents/MP1_Sources/
1|
2 ENTRY(start)
3 phys = 0x00100000;
4 SECTIONS
5 {
6     .text phys : AT(phys) {
7         code = .;
8         *(.text)
9         *(.gnu.linkonce.t.*)
10        *(.gnu.linkonce.r.*)
11        *(.rodata)
12        . = ALIGN(4096);
13    }
14    .data : AT(phys + (data - code))
15    {
16        data = .;
17        *(.data)
18        start_ctors = .;
19        *(.ctor*)
20        end_ctors = .;
```

8. Now we look for any references to kernel.bin and change the name of kernel.bin file in makefile and in script file copykernel to kernel.elf.

9. gdb is used to debug the Bochs Environment, where the Bochs emulator. For this to work, we need to enable the gdb stub in the Bochs configuration file. This can be done by adding the following line in bochsrc.bxrc file as shown:

```
gdbstub: enabled=1, port=1234, text_base=0, data_base=0, bss_base=0
```

10. Then we run the commands:

```
Sudo make
Sudo make install
./copykernel.sh
```

11. Loading Bochs, I run the command:

```
bochs -f bochsrc.bxrc
```

12. To connect from GDB, open a new terminal and run:

```
gdb YOUR-KERNEL where YOUR-KERNEL is the ELF output file.
```

We run the commands

```
gdb kernel.elf
(gdb) target set architecture i386
(gdb) target remote 127.0.0.1:1234
```

```
csce410@COE-VM-CSE1-L37: ~/Documents/bochs-2.6.8          csce410@COE-VM-CSE1-L37: ~/Documents/MP1_Sources/MP1_Sources
000000000001[      ] installing x module as the Bochs GUI
000000000001[      ] using log file bochsrc.out
=====
Bochs is exiting with the following message:
[MEM0 ] ROM: couldn't open ROM image file 'VGABIOS-lGPL-latest'.
=====
csce410@COE-VM-CSE1-L37:~/Documents/bochs-2.6.8$ bochs -f bochsrc.bxrc
=====
Bochs x86 Emulator 2.6.8
Built from SVN snapshot on May 3, 2015
Compiled on Jan 28 2024 at 17:33:39
=====
000000000001[      ] BXSHARE not set. using compile time default '/usr/local/share/bochs'
000000000001[      ] reading configuration from bochsrc.bxrc
000000000001[      ] Enabled gdbstub
=====
Bochs Configuration: Main Menu
-----

This is the Bochs Configuration Interface, where you can describe the
machine that you want to simulate. Bochs has already searched for a
configuration file (typically called bochsrc.txt) and loaded it if it
could be found. When you are satisfied with the configuration, go
ahead and start the simulation.

You can also start bochs with the -q option to skip these menus.

1. Restore factory default configuration
2. Read options from...
3. Edit options
4. Save options to...
5. Restore the Bochs state from...
6. Begin simulation
7. Quit now

Please choose one: [6] 6
000000000001[      ] installing x module as the Bochs GUI
000000000001[      ] using log file bochsrc.out
Waiting for gdb connection on port 1234
Connected to 127.0.0.1
[ Screenshot ]
```

```
csce410@COE-VM-CSE1-L37: ~/Documents/MP1_Sources/MP1_Sources          csce410@COE-VM-CSE1-L37: ~/Documents/MP1_Sources/MP1_Sources
Recents
Star Start
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from kernel.elf...
(gdb) set architecture i386:x86-64:intel
The target architecture is assumed to be i386:x86-64:intel
(gdb) target remote localhost:1234
localhost:1234: connection timed out.
(gdb) target remote localhost:1234
Remote debugging using localhost:1234
(gdb) PC register is not available
(gdb) set architecture i386
(gdb) The target architecture is assumed to be i386
(gdb) target remote localhost:1234
localhost:1234: Connection timed out.
(gdb) quit
csce410@COE-VM-CSE1-L37:~/Documents/MP1_Sources/MP1_Sources$ gdb kernel.elf
GNU gdb (Ubuntu 9.2-0ubuntu1-20.04.1) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from kernel.elf...
(gdb) set architecture i386
The target architecture is assumed to be i386
(gdb) target remote 127.0.0.1:1234
127.0.0.1:1234: connection timed out.
(gdb) target remote 127.0.0.1:1234
Remote debugging using 127.0.0.1:1234
0x0000ffff in ?? ()
(gdb) [ Screenshot ]
```

The screenshot shows a terminal window titled "Bochs Configuration Interface" with two tabs open. The left tab shows the configuration file content:

```
Bochs is exiting with the following message:  
[ ] bochsrc.bxrc:16: a bochsrc option needs at least one parameter  
=====  
Bochs x86 Emulator 2.6.8  
Built from SVN snapshot on May 3, 2015  
Compiled on Jan 28 2024 at 17:33:39  
BXSHARE not set. using compile time default '/usr/local/share/bochs'  
reading configuration from bochsrc.bxrc  
Enabled gdbstub  
=====  
Bochs Configuration: Main Menu  
+ Options  
tsd This is the Bochs Configuration Interface, where you can describe the machine that you want to simulate. Bochs has already searched for a configuration file (typically called bochsrc.txt) and loaded it if it could be found. When you are satisfied with the configuration, go ahead and start the simulation.  
+ Other  
You can also start bochs with the -q option to skip these menus.  
1. Restore factory default configuration  
2. Read options from...  
3. Edit options  
4. Save options to...  
5. Restore the Bochs state from...  
6. Begin simulation  
7. Quit now  
Please choose one: [6] 6  
000000000001[ ] installing x module as the Bochs GUI  
000000000001[ ] using log file bochcout.txt  
Waiting for gdb connection on port 1234  
Connected to 127.0.0.1
```

The right tab shows the command history:

```
csce410@COE-VM-CSE1-L37: ~/Documents/MP1_Sources/MP1_Sources$ subl bochsrc.bxrc  
csce410@COE-VM-CSE1-L37: ~/Documents/MP1_Sources/MP1_Sources$ bochs -f bochsrc.bxrc
```

Finally running commands

(gdb) b main()

(gdb) continue

The screenshot shows a Bochs x86 emulator interface. At the top, there's a toolbar with icons for A1, B1, CD, and floppy disk, along with options like USER, Copy, Paste, Snapshot, Reset/Suspend/Power, and CONFIG. Below the toolbar, a message says "Initialized console. Replace the following <NAME> field with your name. After you are done admiring your output, you can shutdown this 'machine'." The terminal window displays "WELCOME TO MY KERNEL!" followed by "KGALIANGI SINHA?" in red. In the bottom right corner of the terminal, there's a "Screenshot" button. The background of the terminal window is dark purple.

```
Bochs x86 emulator, http://bochs.sourceforge.net/
Initialized console.
Replace the following <NAME> field with your name.
After you are done admiring your output, you can shutdown this 'machine'.
WELCOME TO MY KERNEL!
KGALIANGI SINHA?

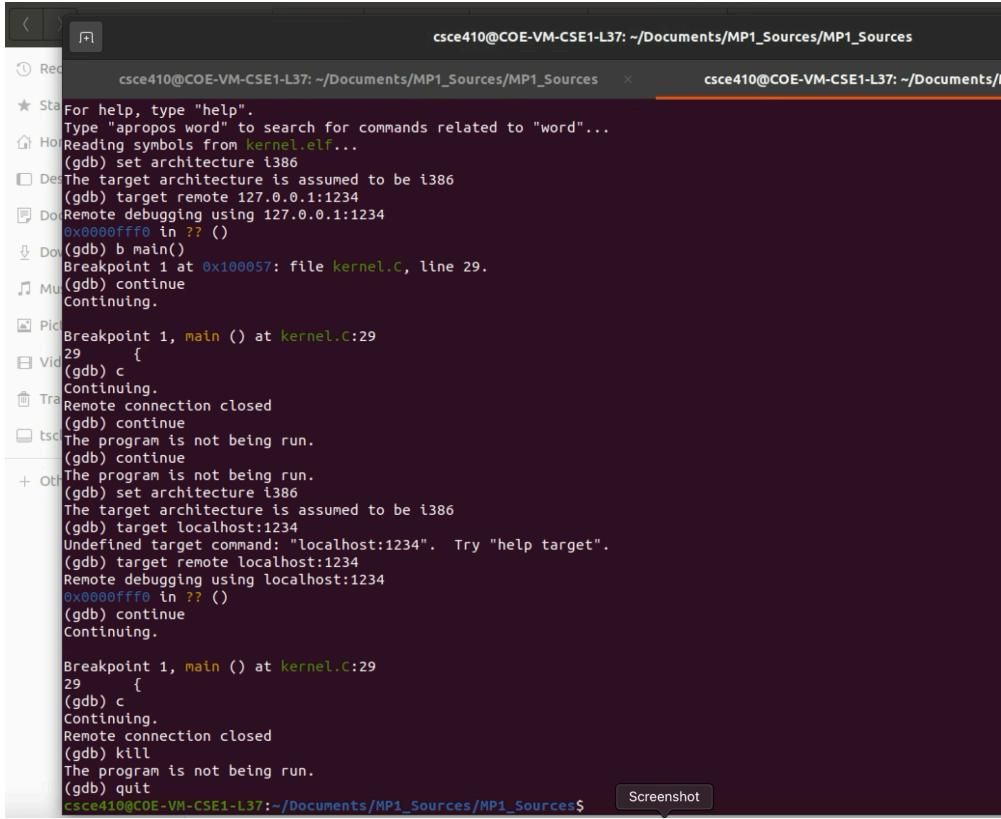
lf
tml>

IPS: 5.66GM      A: NUM  CAPS  SCRL
+ Other  <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from kernel.elf...
(gdb) set architecture i386
The target architecture is assumed to be i386
(gdb) target remote 127.0.0.1:1234
Remote debugging using 127.0.0.1:1234
0x0000ffff in ?? ()
(gdb) b main()
Breakpoint 1 at 0x100057: file kernel.C, line 29.
(gdb) continue
Continuing.

Breakpoint 1, main () at kernel.C:29
29  {
(gdb) c
Continuing.
Screenshot
```

You can end the debugging session by running the kill (k) command to terminate the debugging process and exit GDB with the quit (q) command.



The screenshot shows a terminal window with two tabs open. Both tabs have the title "csce410@COE-VM-CSE1-L37: ~/Documents/MP1_Sources/MP1_Sources". The left tab displays a GDB session log:

```
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from kernel.elf...
(gdb) set architecture i386
The target architecture is assumed to be i386
(gdb) target remote 127.0.0.1:1234
Remote debugging using 127.0.0.1:1234
0x0000ffff in ?? ()
(gdb) b main()
Breakpoint 1 at 0x100057: file kernel.c, line 29.
(gdb) continue
Continuing.

Breakpoint 1, main () at kernel.c:29
29  {
(gdb) c
Continuing.
Remote connection closed
(gdb) continue
The program is not being run.
(gdb) continue
Continuing.

Breakpoint 1, main () at kernel.c:29
29  {
(gdb) c
Continuing.
Remote connection closed
(gdb) kill
The program is not being run.
(gdb) quit
```

The right tab shows a blank command line interface.