

Integration gdb with bochs

Mian Qin UIN:725006574

1 recompile bochs with gdb-stub

From the first helper link, I recompile bochs with version 2.6.8 with `--enable-gdb-stub --with-x11`. This is mandatory.

During the compile of bochs, I need to install libraries

```
$ sudo apt-get install libx11-dev
```

```
$ sudo apt-get install libxrandr-dev
```

2 modified make file with `-g` and `-O0` options

Add `-g` and `-O0` options to the `GCC_OPTIONS` in makefile as follow

```
GCC_OPTIONS = -m32 -nostdlib -fno-builtin -nostartfiles -nodefaultlibs -fno-exceptions -  
fno-rtti -fno-stack-protector -fleading-underscore -fno-asynchronous-unwind-tables -g -  
O0
```

3 modified linker.ld to generate elf format binary

Remove the first line of linker.ld

From the second helper link

4 add gdb configuration to the bochs configuration file

In `bochsrc.bxrc`

Add the following lines in the end of file

```
# gdb
```

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

5 write a `gdbinit` file in the MP1 source directory

file `kernel.bin`

```
target remote:1234
```

6 recompile the project and run with bochs

```
$ make
```

```
$ ./copykernel.sh
```

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

7 run gdb to enable remote debugging

```
$ gdb
```

Here are the screenshots for the gdb debugging in action

I add to lines in the `kernel.C` to verify the debug

```
36 int i=0;
```

```
37 char j='a';
```

