

实验一：UNIX V6++运行调制环境的安装与配置

1. 实验目的

- (1) 安装配置 UNIX V6++的运行环境；
- (2) 安装配置 UNIX V6++的调试环境。

2. 实验设备及工具

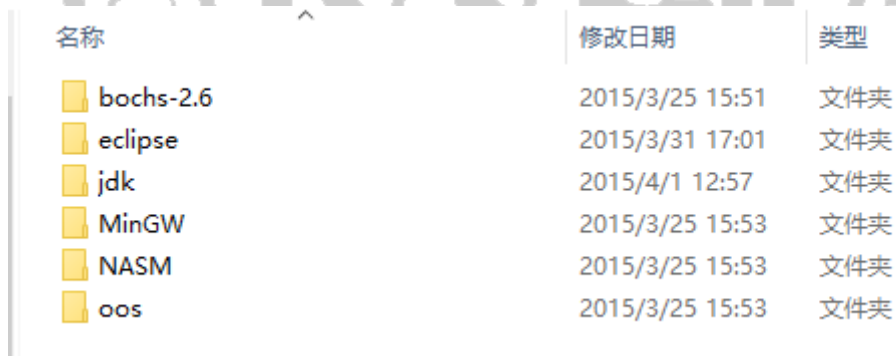
已安装 Windows 操作系统的 PC 机一台。

3. 预备知识

- (1) gcc 编译
- (2) gdb 调试。

4. 实验准备

请根据自己系统的时机情况，选择下载 32 位或者 64 位的工具包。将工具包解压到本地硬盘的任意位置后，出现如图 1 所示的目录结构。



名称	修改日期	类型
bochs-2.6	2015/3/25 15:51	文件夹
eclipse	2015/3/31 17:01	文件夹
jdk	2015/4/1 12:57	文件夹
MinGW	2015/3/25 15:53	文件夹
NASM	2015/3/25 15:53	文件夹
oos	2015/3/25 15:53	文件夹

图 1：UNIX V6++工具包目录结构

bochs-2.6: 一款 X86 硬件平台的开源模拟器，它可以在任何编译运行 bochs 的平台上模拟 x86 硬件，包括：I/O 设备，内存和 BIOS。通过改变配置，可以指定使用的 CPU（386，486 或 586），以及内存大小等等。工具包中提供的是免安装的版本。关于 bochs 更多的细节，可搜索百度百科或查询 bochs 官网：<http://bochs.sourceforge.net/>。

eclipse: 免安装的 eclipse 软件包，用于配置 UNIX V6++的调试环境用。

jdk: Java 虚拟机，运行 eclipse 时需要。

MinGW: 是 Minimalist GNU on Windows 的缩写。它是一个可自由使用和自由发布的 Windows 特定头文件和使用 GNU 工具集导入库的集合, 允许在 GNU/Linux 和 Windows 平台生成本地的 Windows 程序, 而不需要第三方 C 运行时库。这里提供的为免安装版本。关于 MinGW 更多的细节, 可搜索百度百科或查询 MinGW 官网: <http://www.mingw.org/>。

NSAM: 是一个为可移植性与模块化而设计的一个 80x86 的汇编器, 它支持相当多的目标文件格式。这里提供的为免安装版本。关于 NSAM 的更多细节, 可搜索百度百科。

oos: 所有 UNIX 代码所在的文件夹。其中, 所有的源代码位于 oos/src。

5. 实验内容

5.1. UNIX V6++运行环境的配置

在解压后的工具包中, “UNIX V6++\oos\targets\UNIXV6++” 目录下, 有 “bochsrc.bxrc” 和 “c.img” 两个文件 (如图 2 所示)。

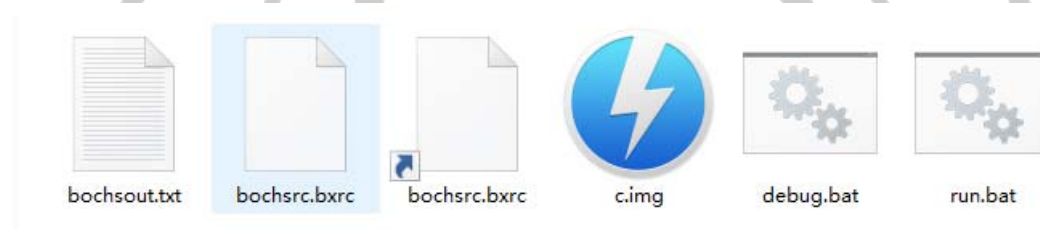


图 2: “UNIX V6++\oos\targets\UNIXV6++” 目录下的文件内容

其中, “bochsrc.bxrc” 是 bochs 的配置文件, 里面配置了 UNIX V6++运行需要的物理环境, 比如 32M 内存等, 图 3 示出了该文件的主要内容。在不影响 UNIX V6++的运行和调试的基础上, 读者可尝试对 “bochsrc.bxrc” 做小的改动。比如, 可将其中 “floppya” 和 “floppyb” 表示两块 UNIX V6++并未使用到的软盘注释掉。

这里需要注意的是: “bochsrc.bxrc” 文件中的第一行:

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

请通过加 “#” 将其注释掉, 这一条语句是用来控制 bochs 虚拟机的 gdb 调试功能的, 我们将在下一个实验内容中详细解释。

“c.img” 是 UNIX V6++的磁盘镜像文件。UNIX V6++文件系统中包含的内容对应于 “UNIX V6++\oos\tools\MakeImage\bin\Debug\programs” 文件夹下的内容。执行 all 命令后 programs 文件夹下的内容会被写到 UNIX V6++的磁盘文件 c.img。

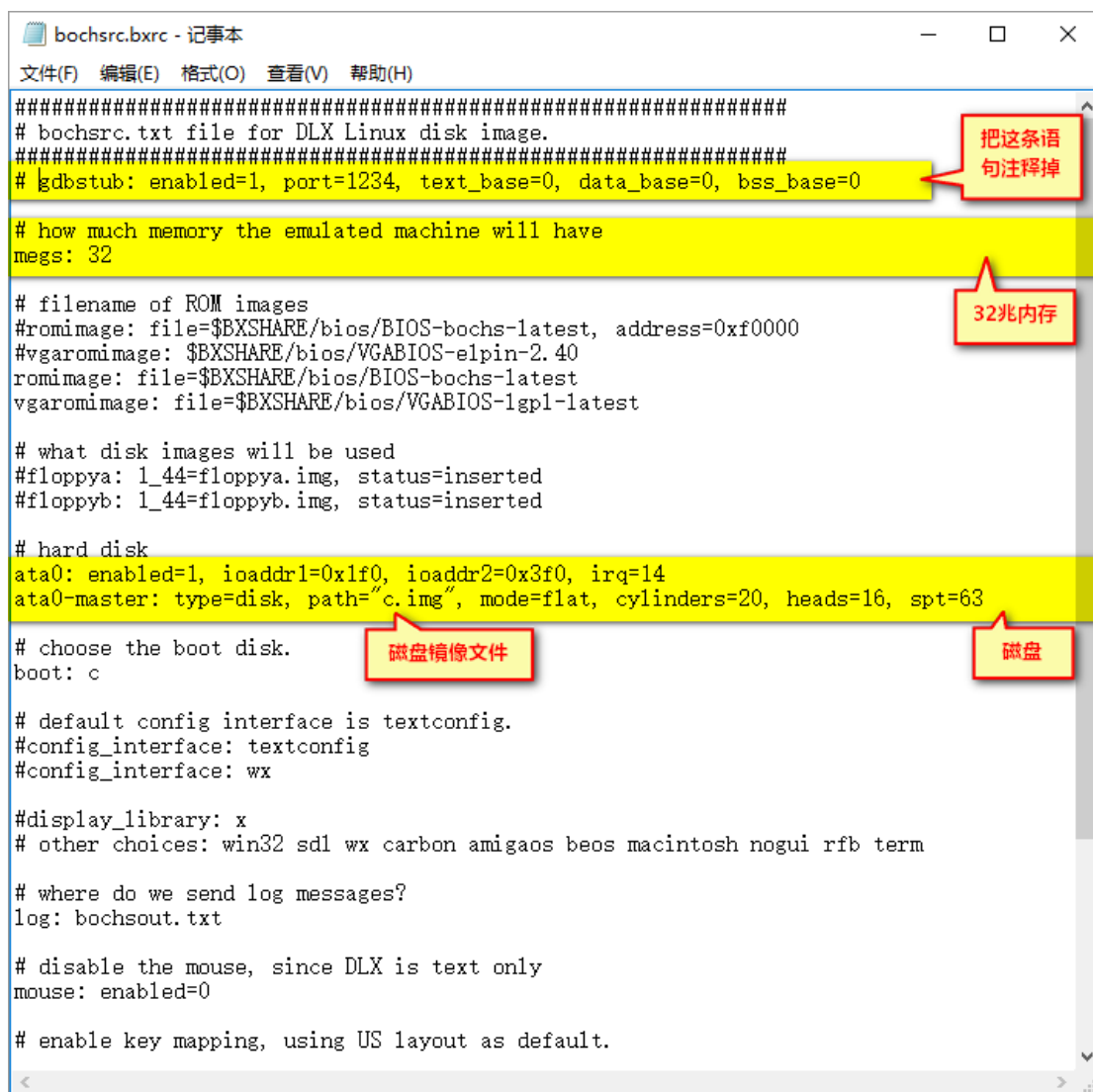


图 3: “bochsrc.bxrc”文件基本结构

为了 UNIX V6++ 的运行，需要完成一些环境变量的设置，在 “UNIX V6++\oos\tools\” 目录下，有一个名为 “oosvars_mingw.bat” 的文件（如图 4 所示）。用文本编辑器打开该文件后，可以看到其中关于 bochs, MinGW 及 NSAM 等工具所在路径的设置。把它们修改成这些文件在你的机器中的正确路径（如图 5 所示）。

5.2. 运行 UNIX V6++

完成了上面的各项参数设置之后，运行 “UNIX V6++\oos\tools\” 目录下的 OOS Command Prompt 快捷方式。运行后弹出如图 6 所示的命令行窗口，在其中输入 “run” 后，首先启动 bochs 虚拟机，然后启动 UNIX V6++（如图 7 所示）。



图 5: oosvars_mingw.bat 文件内容

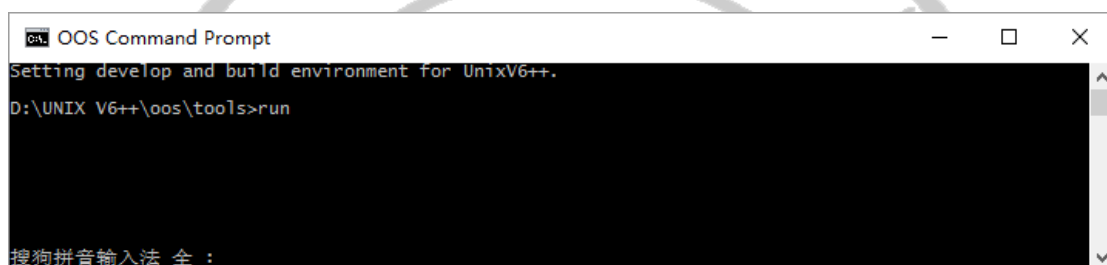


图 6: 通过命令行方式启动 UNIX V6++

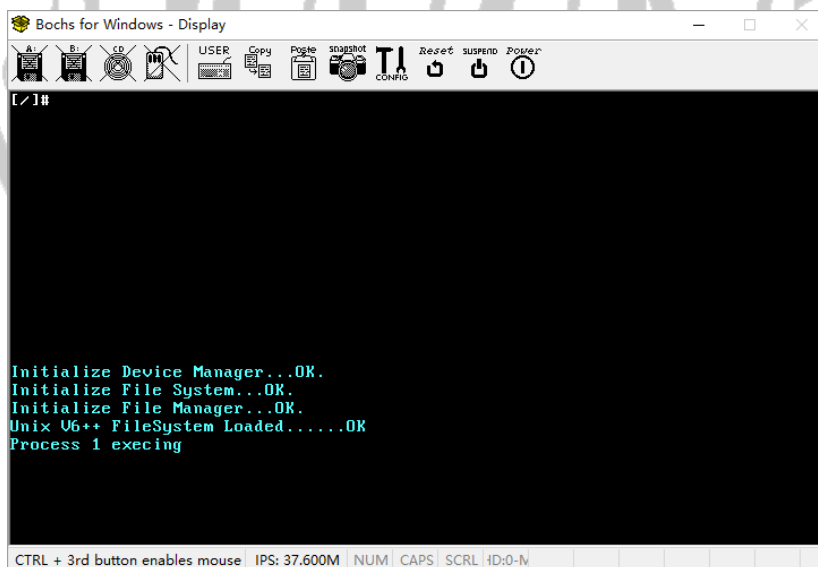


图 7: UNIX V6++运行界面

(3) 使用 UNIX V6++

在启动的 UNIX V6++终端上输入 shell 命令。UNIX V6++中只实现了很少的几个 shell 命令，可通过图 8 中所示的操作过程查看 UNIX V6++中支持的命令。

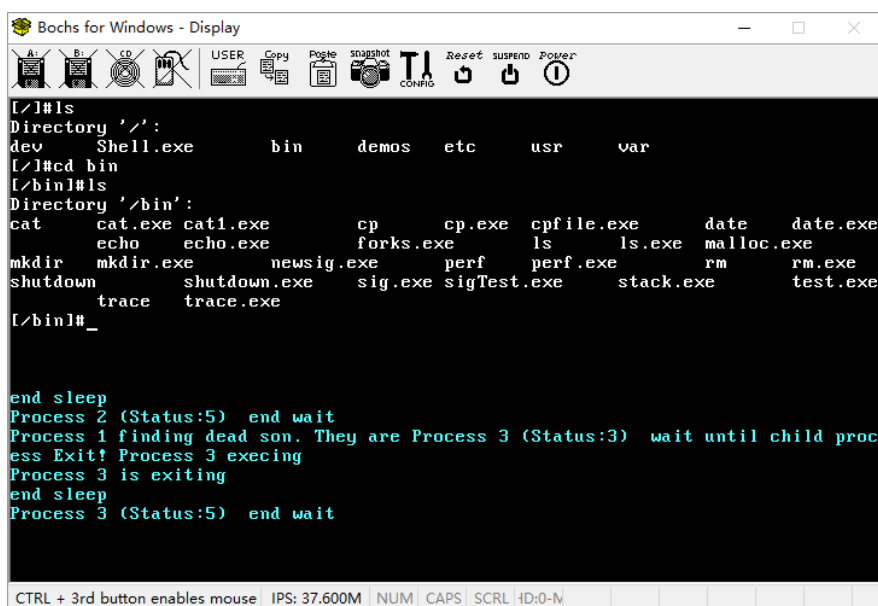


图 8: UNIX V6++运行效果

为了更好的完成对 UNIX V6++代码的修改和重新编译,我们需要一个 gdb 的调试环境。这里我们详细介绍这个 gdb 调试环境的建立过程。

5.3. UNIX V6++调试环境的配置

(1) 开启 bochs 虚拟机的调试选项

在实验一中,我们曾经提到过“bochsrc.bxrc”文件中的第一行:

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

是需要被注释掉的,表示这个时候的 bochs 虚拟机是不支持 gdb 调试功能的,所以首先需要将这一句的注释去掉(如图 9 所示)。

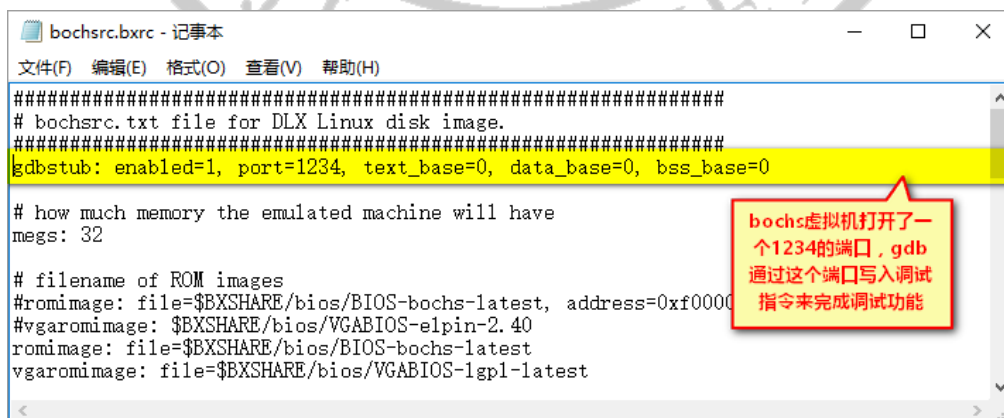


图 9: 开启 gdbstub:注释项

这个时候，我们再次运行 UNIX V6++（注意，如果这时候系统弹出如图 10 所示的对话框，请在勾选“专用网络”之后点击“允许访问”，或者直接点击“取消”），此时，将看到如 11 所示的运行效果。可以看到，虚拟机暂停运行后，正在等待来自 gdb 的调试指令，没有再继续运行下去。

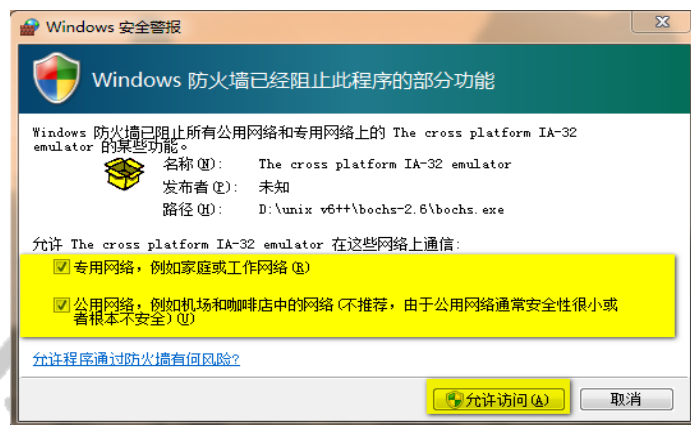


图 10：关闭 Windows 防火前

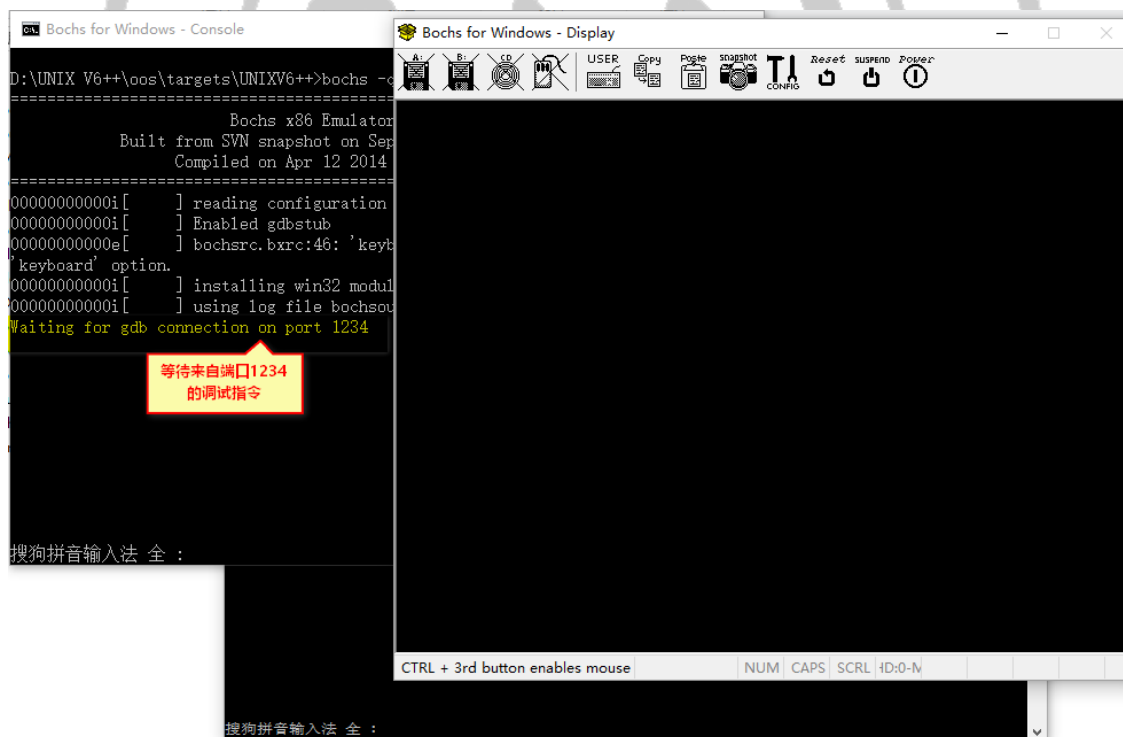


图 11：调试状态运行的 UNIX V6++

至此，熟悉 gdb 调试的同学，可以打开另外一个终端，通过 gdb 调试指令来控制 UNIX V6++ 的指令，完成所有的调试任务。但是，由于 gdb 的调试比较复杂，难于掌握，为了使用的方便，我们将继续步骤，以实现利用 eclipse 提供的图形化的前端来完成 UNIX V6++ 的远程调试。

(2) 调试相关的系统环境变量设置

为了支持 UNIX V6++的调试，需要在 windows 操作系统中设置环境变量，具体的操作过程为：（1）在“高级系统设置”中选择“环境变量”设置，（2）选择其中的“path”变量，通过编辑该变量，将 MinGW 和 NSAM 的路径添加进去（如果该变量不存在，点击“新建”添加）。具体流程如图 12 所示。

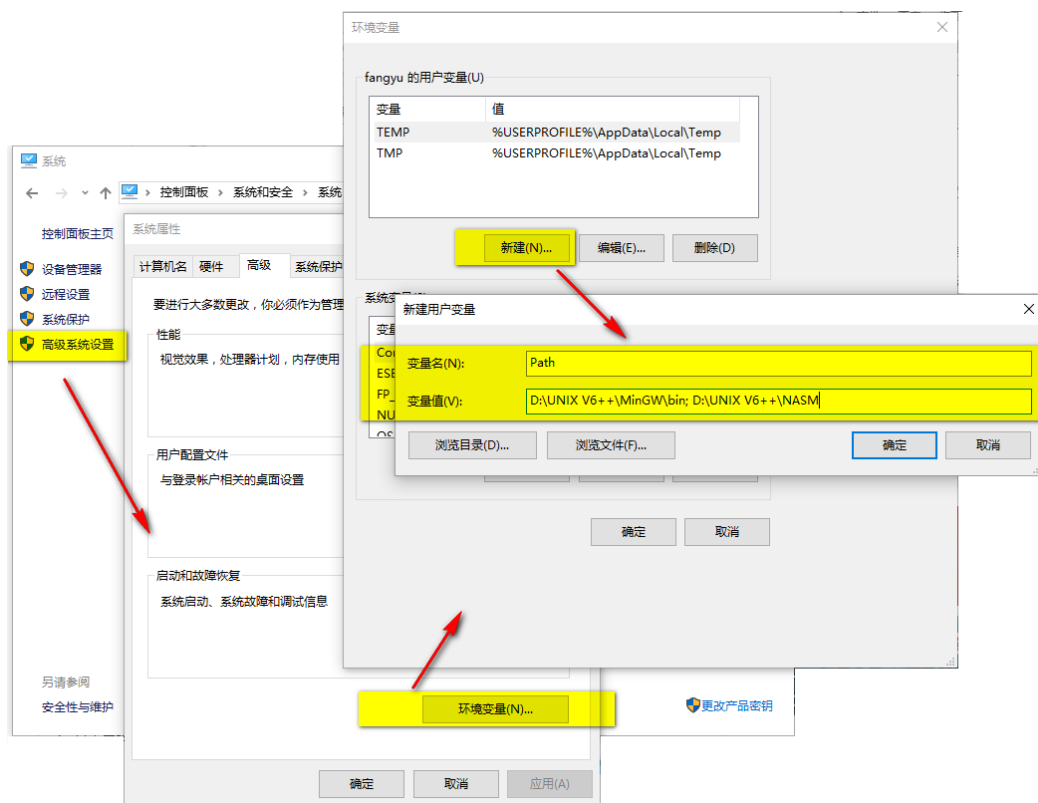


图 12: 系统环境变量设置

5.4. ECLIPS 远程调试环境的配置

(1) 启动 eclips

建议使用工具包中自带的这个版本，如果自行下载和安装的其他版本，有可能导致调试环境配置失败。需要注意的是，安装 `eclipse` 之前，请确保机器中已经安装好 `java` 虚拟机。如果没有安装，可选择工具包中提供的 `java` 虚拟机进行安装。

(2) 将 oos 工程引入到 eclipse 中

通过 File 下的 import 菜单项，将 oos 工程引入到 eclipse 中，操作完成后，可在 eclipse 中看到 oos 工程的全部源代码（如图 13~图 16 所示）。

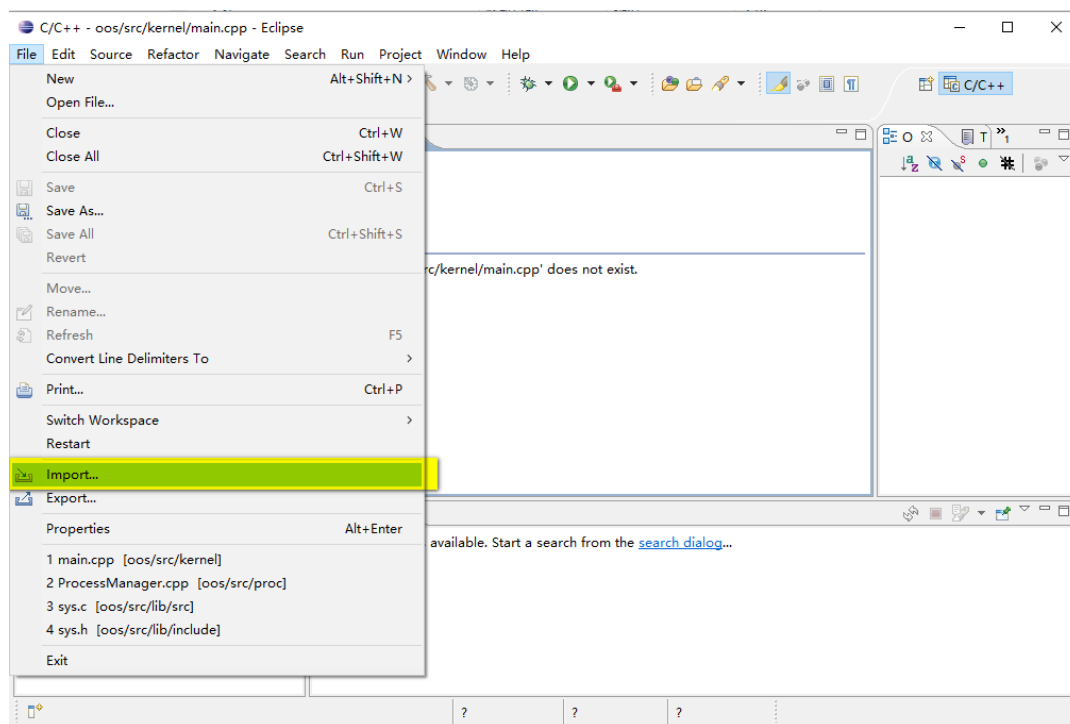


图 13

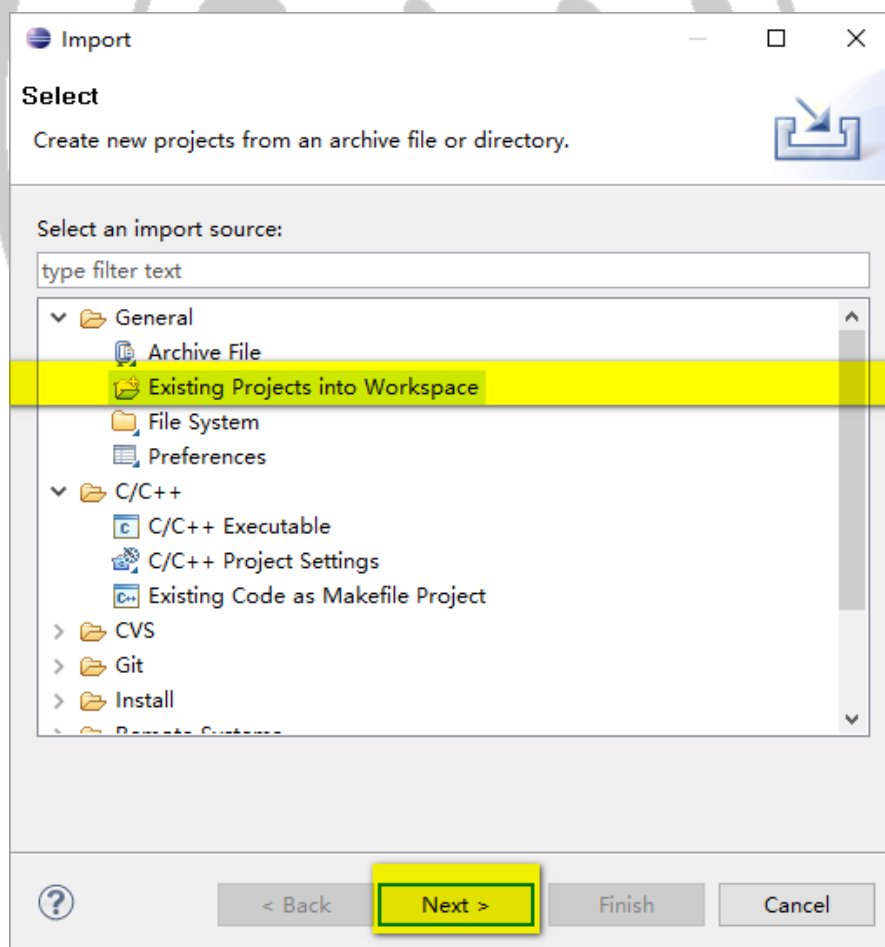


图 14

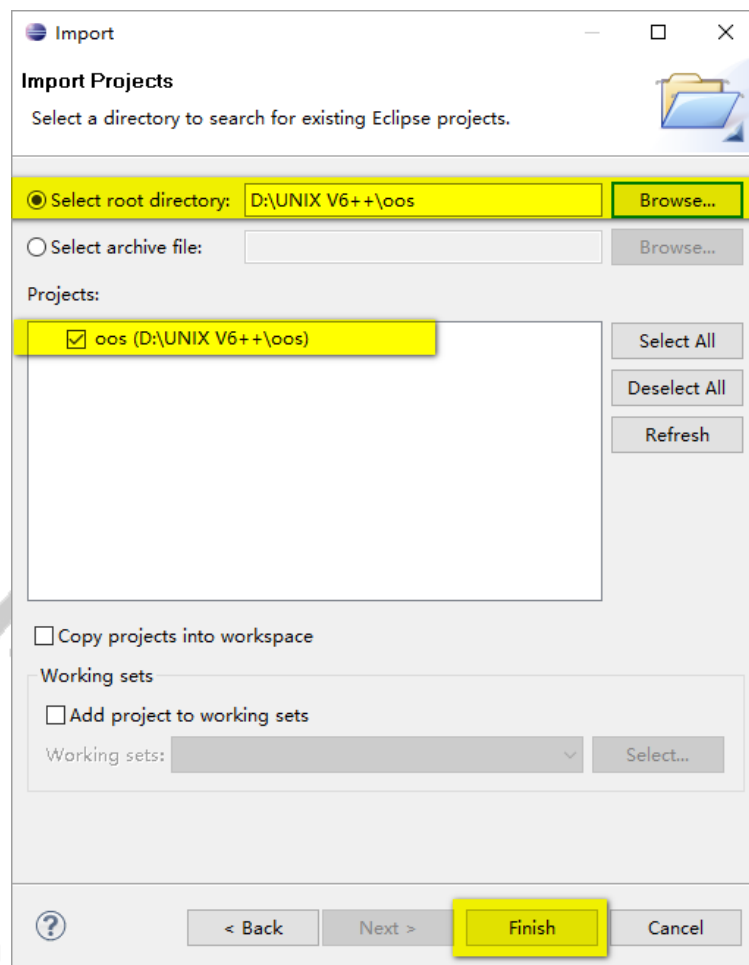


图 15

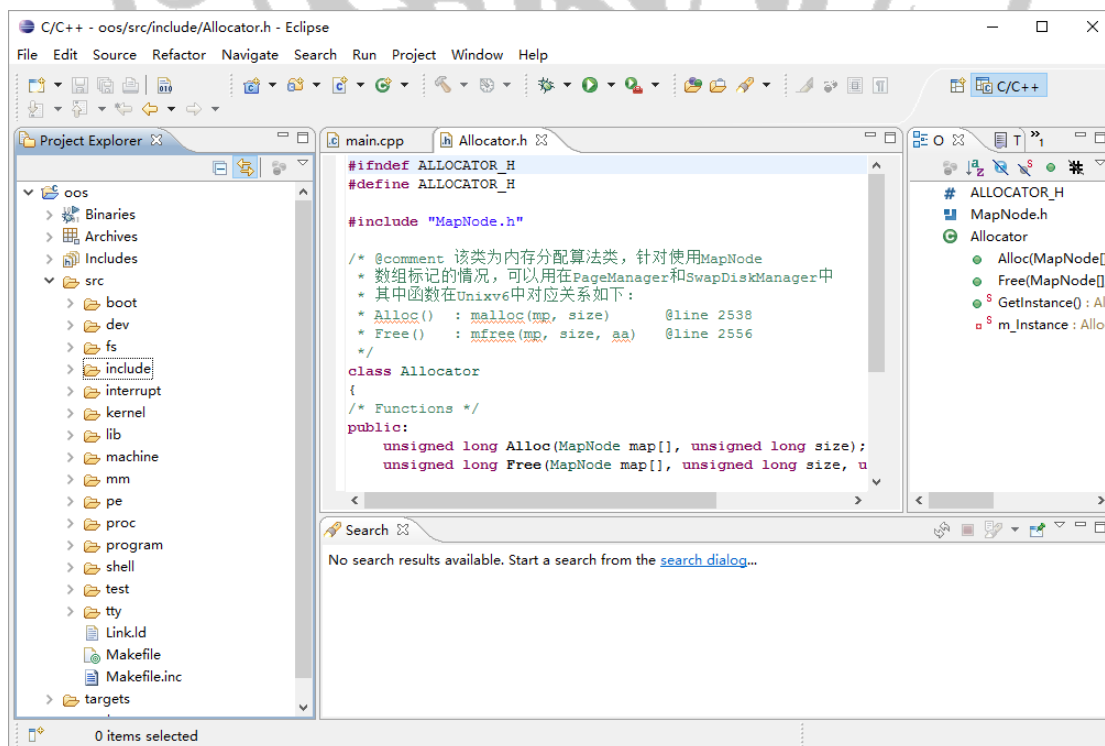


图 16

(3) 设置 oos 工程属性

通过 File 下的 Properties 菜单项，选择“将 C++ Build/tool Chain Editor”，其中的 Current tool Chain 选为 Gnu Make Builder，去掉 Generate Makefiles automatically 选项，设置 build directory 为 oos/src（如图 17~ 图 19 所示）。

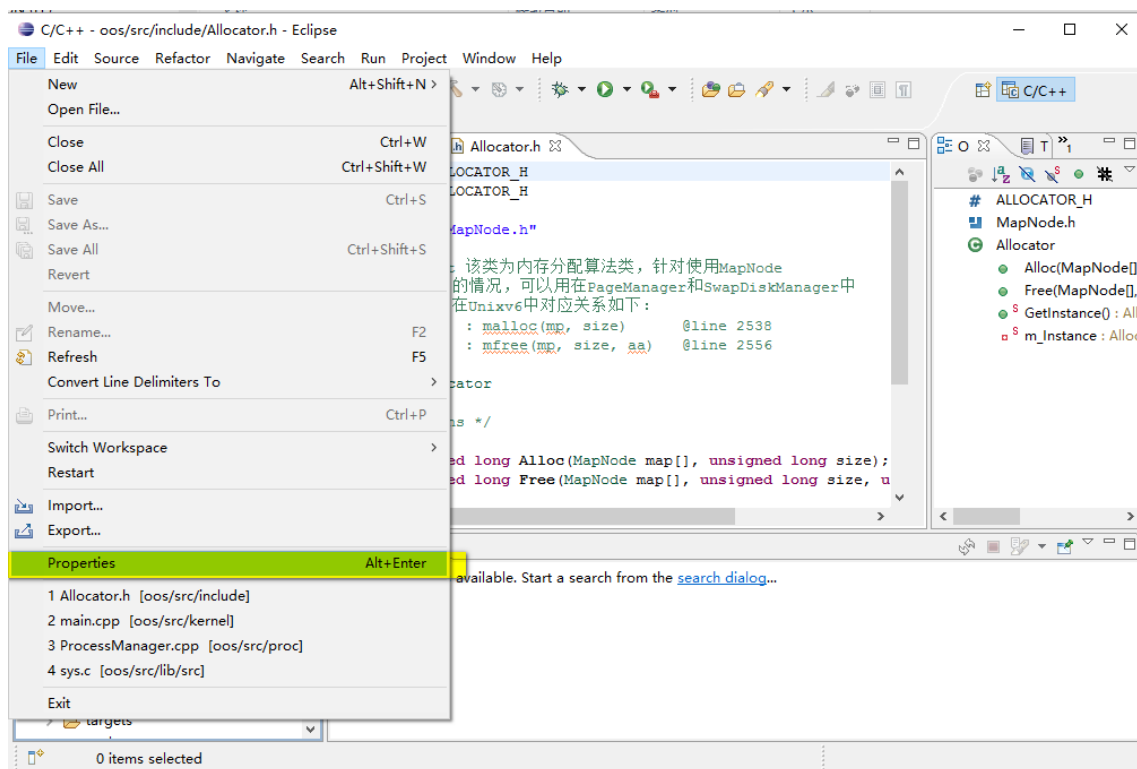


图 17

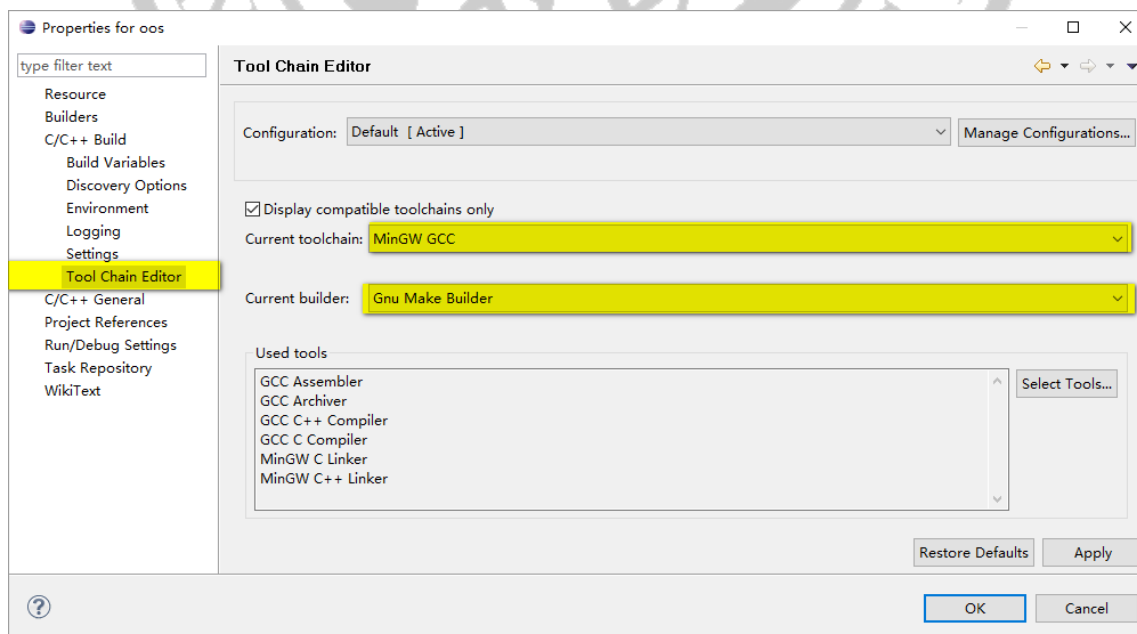


图 18

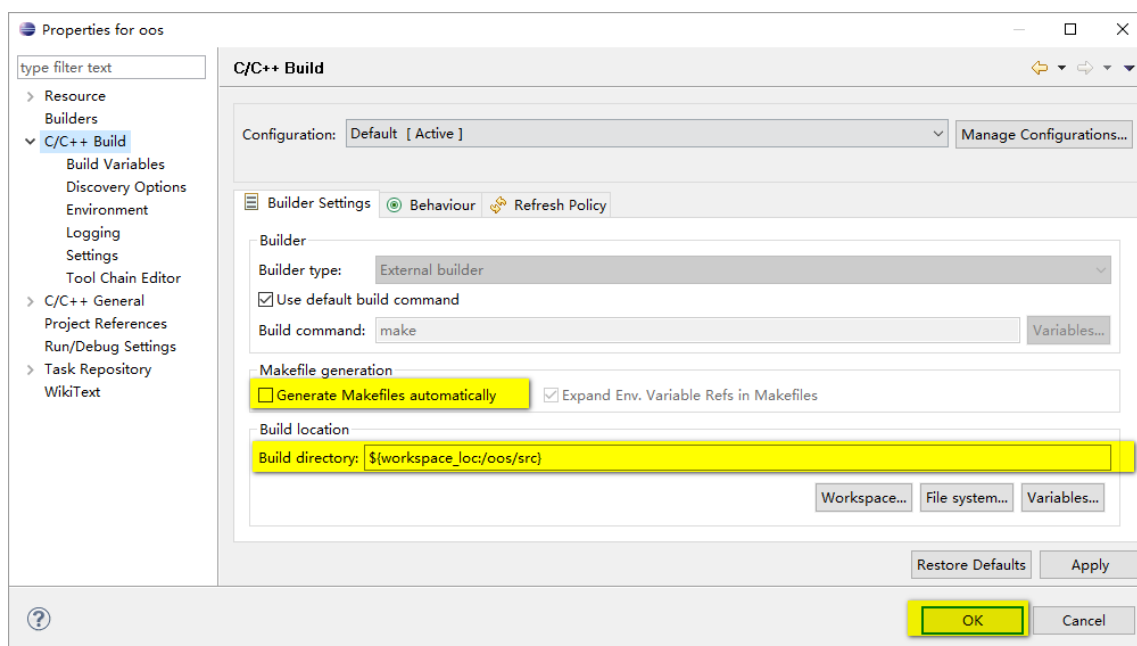


图 19

(3) 配置 gdb 远程调试

配置 gdb 远程调试的过程如下：

- 进入“Debug Configurations”菜单项，新建一个 C/C++ Application 配置（右击“C/C++ Application”选项，单击“new”）
- 起名“oos”，设置调试对象为 UNIX V6++的 kernel.exe 文件。点选 SelectOther，设置为 Standard Create Process Launcher 方式。
- 在“Debugger”选项卡中，设置调试器为“gdbserver”，起始调试点为“main0”函数，“connection”中设置连接方式为“TCP”，端口号为“1234”，与 bochs 虚拟机中的设置一致。

上述过程参见图 20~24。

至此，我们完成了全部基于 eclipse 和 gdb 的远程调试环境的配置。

5.5. 开始调试

有了 eclipse 的远程调试环境之后，我们可以方便地在图形界面中完成对 UNIX V6++代码的调试工作。具体过程如下：

首先，可以在源程序中的任意位置设置断点，见图 25 中 1。然后，如图 11 所示，启动调试状态的 UNIX V6++。最后，在 eclipse 中开启调试，见图 25 中 2。这时，会看到，UNIX

V6++运行到了断点位置不再执行（如图 26）。此时，在 `eclipse` 中可以查看各个变量的值（如图 25 中 3），或者通过“Registers”标签查看各个寄存器的值。

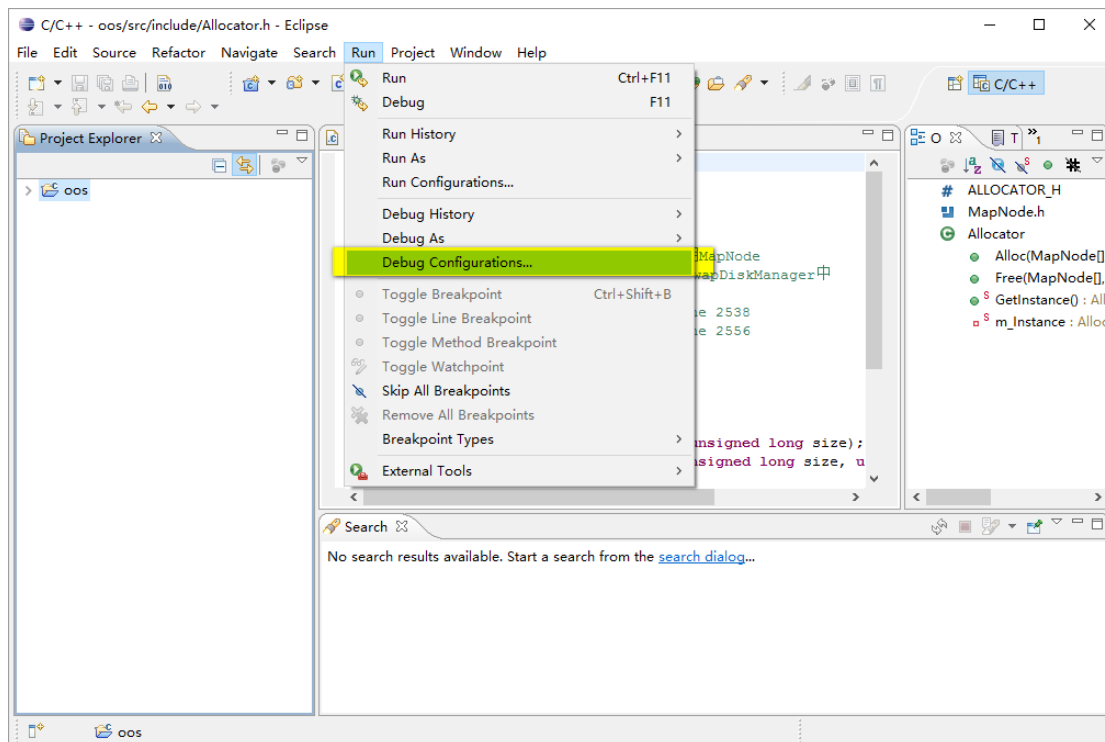


图 20

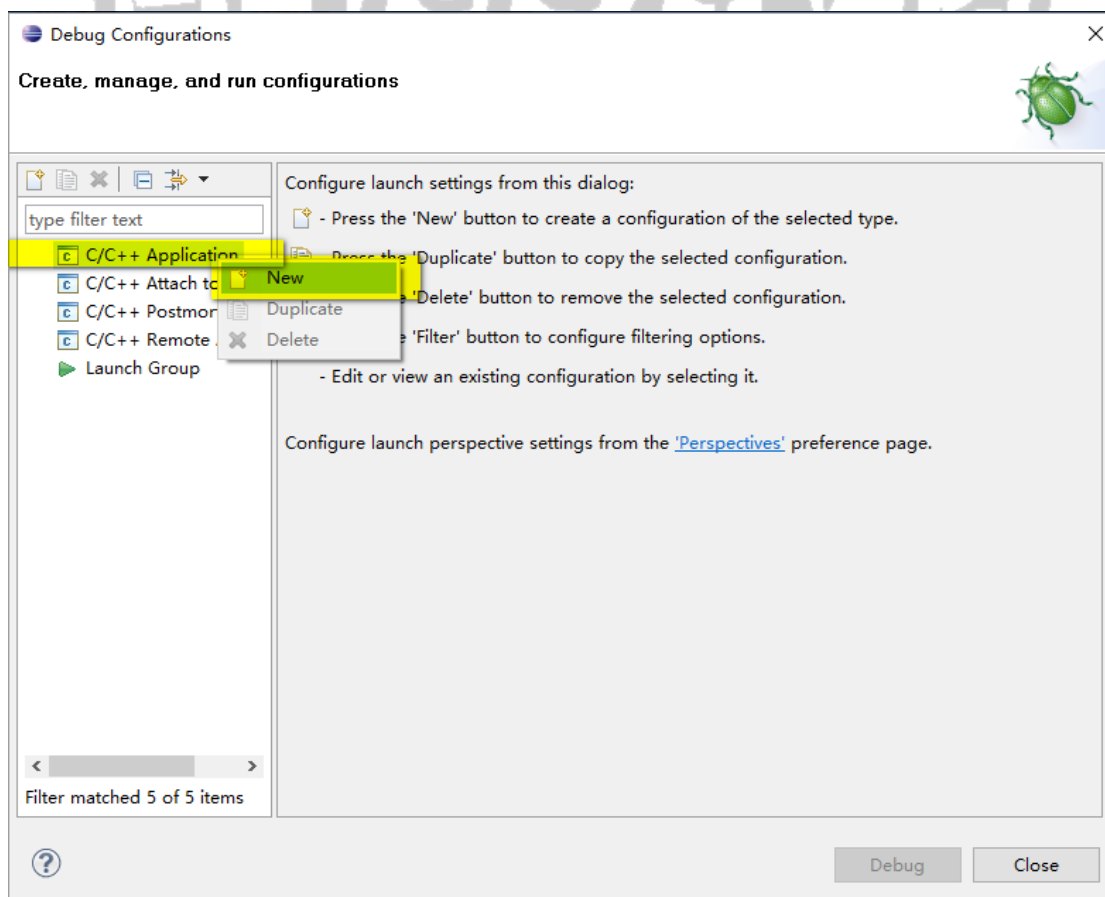


图 21

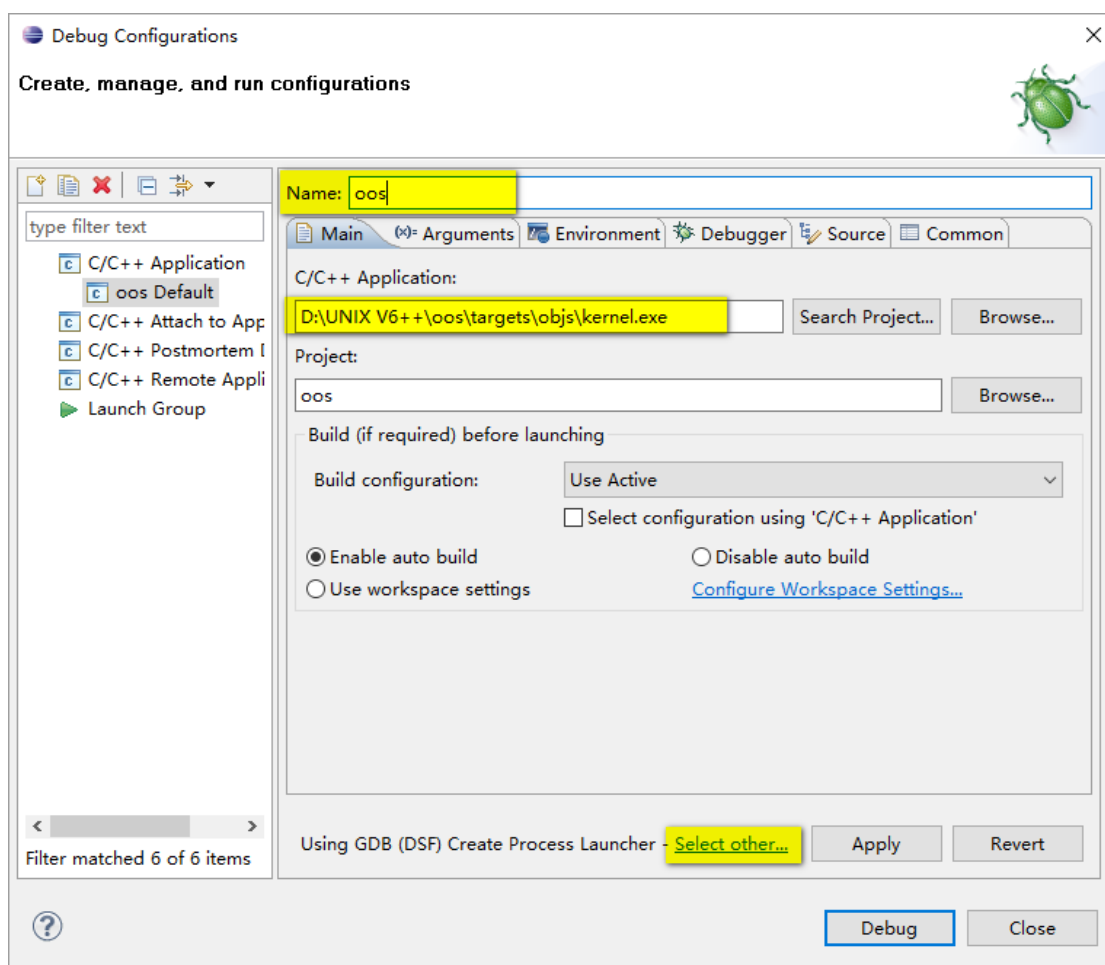


图 22

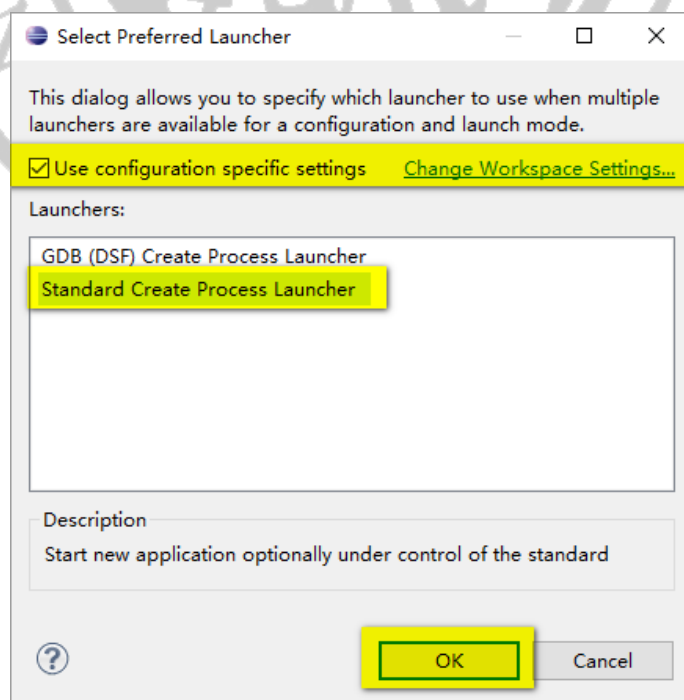


图 23

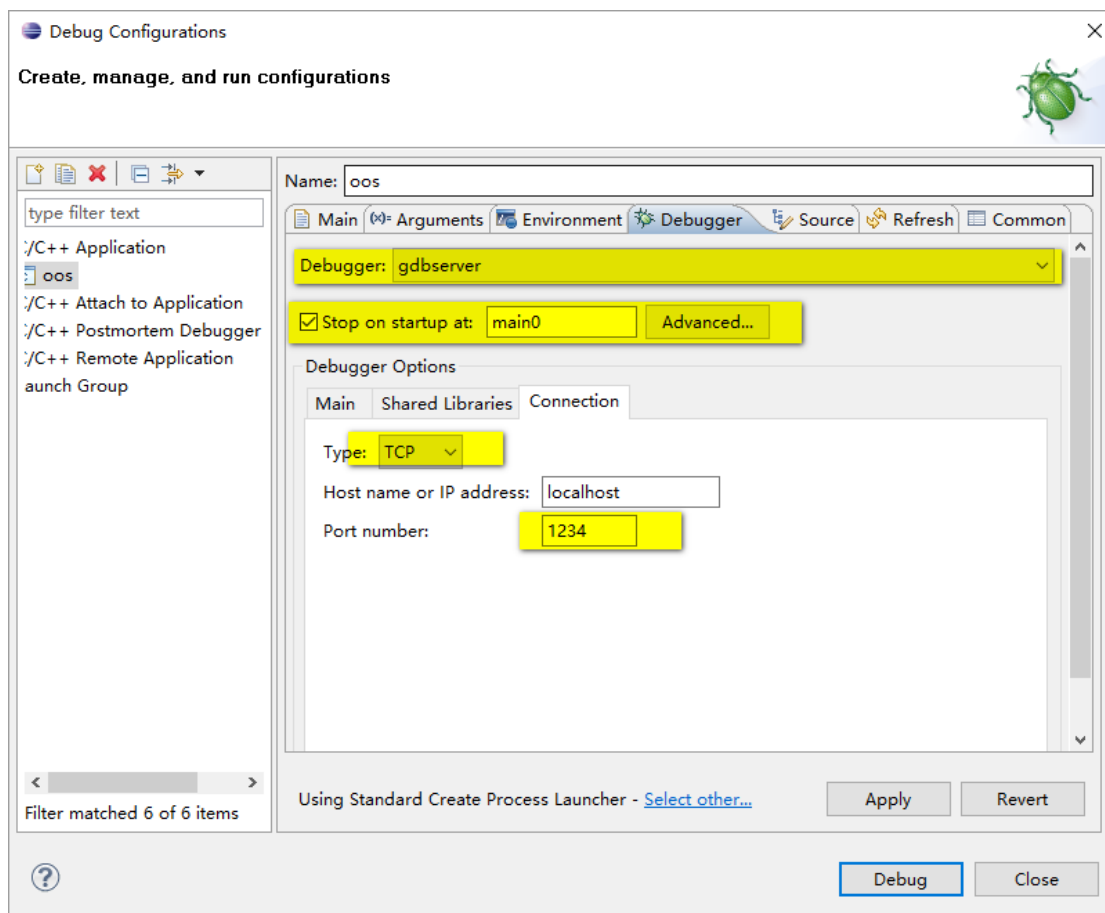


图 24

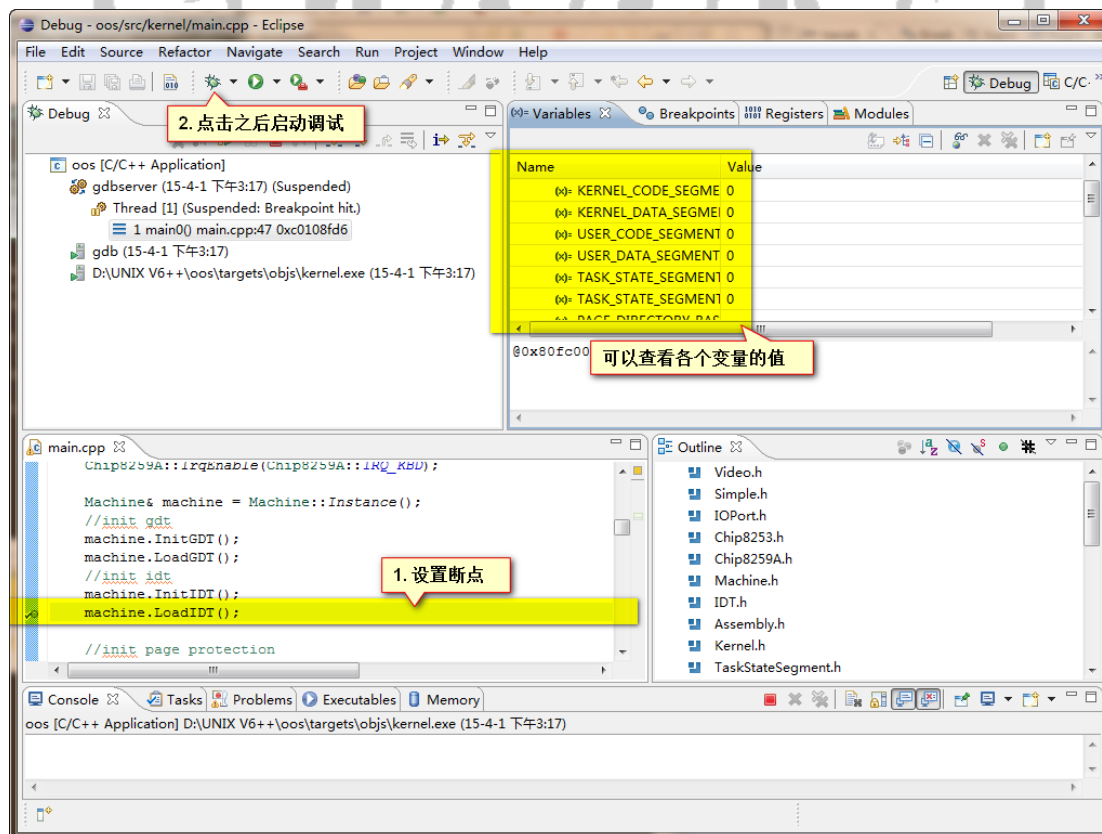


图 25

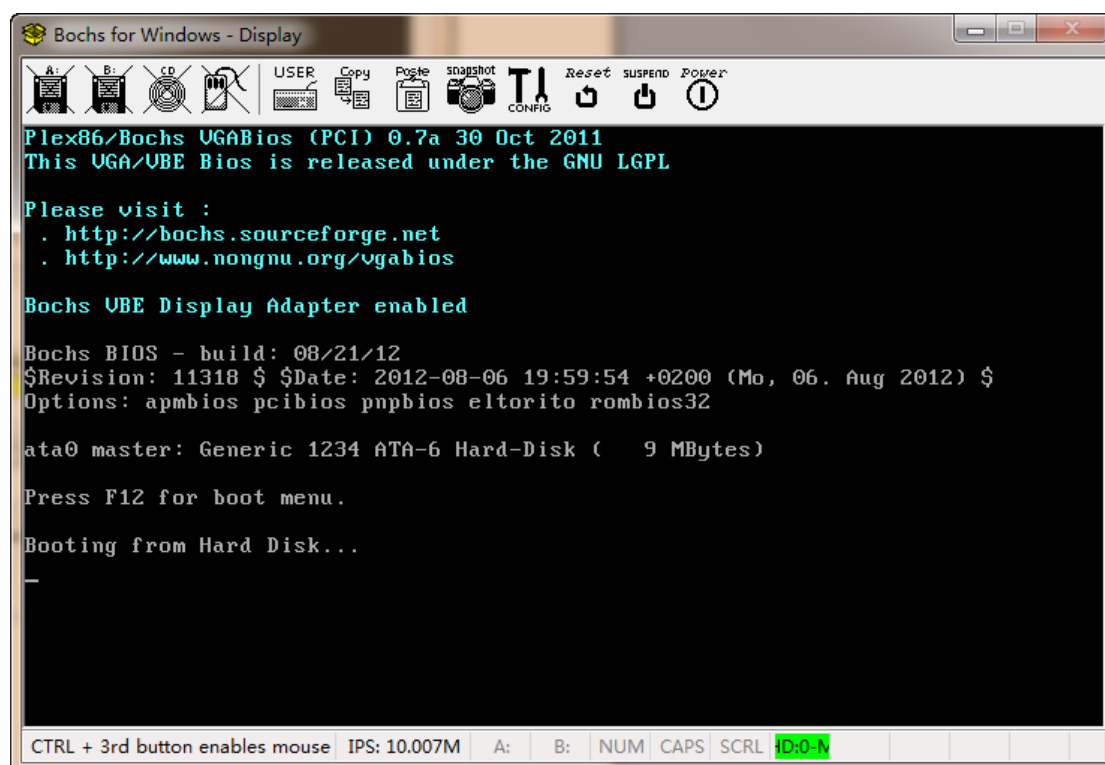


图 26

6. 思考题*

- (1) 在 eclipse 中观察整个 UNIX V6++源代码目录结构, 根据你的理解, 尝试给出每一个 oos/src/下的子目录中所包含文件的用途。
- (2) 通过调试环境的配置和运行, 谈谈你对远程调试的理解。UNIX V6++采用基于 eclipse 远程调试的好处和可能存在的问题有哪些?

7. 实验报告要求

- (1) (2 分) 依照实验指导过程, 完成 UNIX V6++的运行环境的安装与配置, 执行几条简单的 UNIX 指令, 关键步骤截图说明。
- (2) (2 分) 依照实验指导过程, 完成 UNIX V6++的 eclipse 远程调试环境的安装与配置, 关键步骤截图说明; 通过自己设置断点, 观察 UNIX V6++的调试运行, 查看变量与寄存器的值, 截图说明。