

同济大学计算机系

操作系统课程实验报告



学 号 2154312

姓 名 郑博远

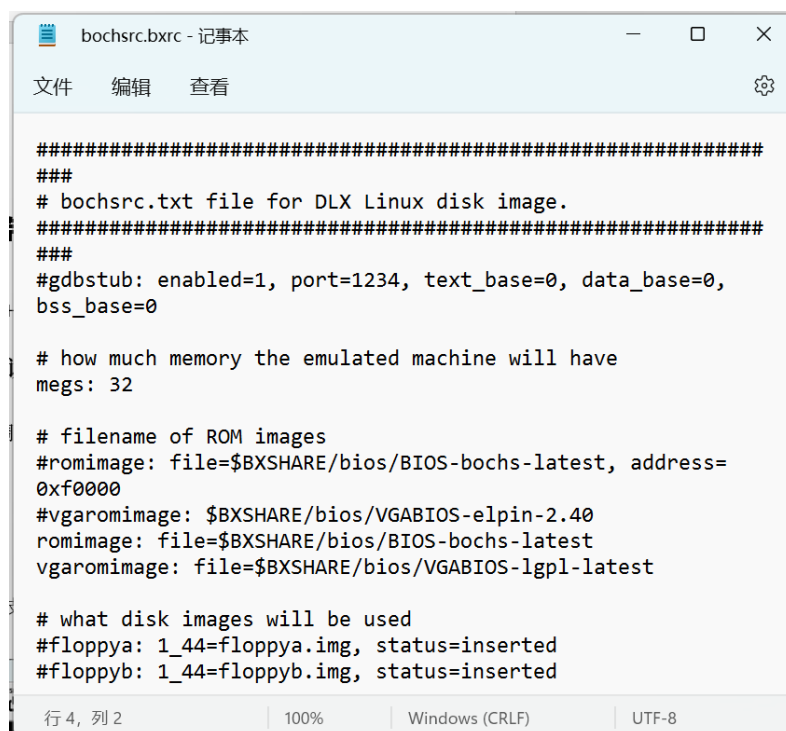
专 业 计算机科学与技术

授课老师 方 钰

P01: UNIX V6++ 运行调制环境的安装与配置

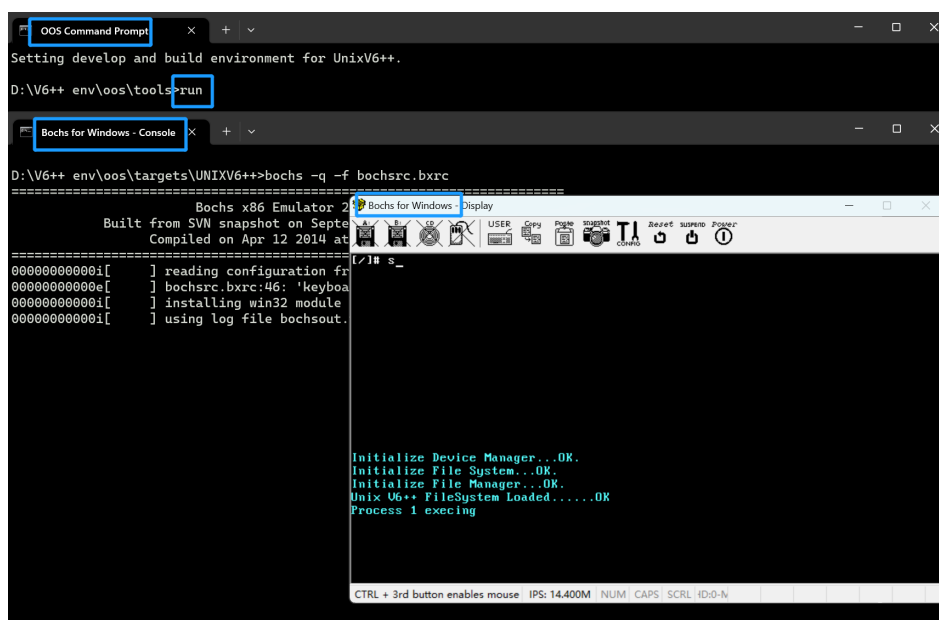
一、 依照实验指导过程，完成 UNIX V6++的运行环境的安装与配置，执行几条简单的 UNIX 指令，关键步骤截图说明。

1. 在“bochsrc.bxrc”文件中注释掉用于调试的语句：



```
#####  
###  
# bochsrc.txt file for DLX Linux disk image.  
#####  
###  
#gdbstub: enabled=1, port=1234, text_base=0, data_base=0,  
bss_base=0  
  
# how much memory the emulated machine will have  
megs: 32  
  
# filename of ROM images  
#romimage: file=$BXSHARE/bios/BIOS-bochs-latest, address=  
0xf0000  
#vgaromimage: $BXSHARE/bios/VGABIOS-elpin-2.40  
romimage: file=$BXSHARE/bios/BIOS-bochs-latest  
vgaromimage: file=$BXSHARE/bios/VGABIOS-lgpl-latest  
  
# what disk images will be used  
#floppya: 1_44=floppya.img, status=inserted  
#floppyb: 1_44=floppyb.img, status=inserted  
  
行 4, 列 2      100%      Windows (CRLF)      UTF-8
```

2. 运行“OOS Command Prompt”



```
OOS Command Prompt  x + v  
Setting develop and build environment for UnixV6++.  
D:\V6++ env\oos\tools\run  
  
Bochs for Windows - Console  x + v  
D:\V6++ env\oos\targets\UNIXV6++>bochs -q -f bochsrc.bxrc  
===== Bochs x86 Emulator 2.6.0 =====  
Built from SVN snapshot on Sept 12 2014 at 14:00:00  
Compiled on Apr 12 2014 at 14:00:00  
===== s- =====  
0000000000i[ ] reading configuration file bochsrc.bxrc:46: 'keyboa  
0000000000e[ ] bochsrc.bxrc:46: 'keyboa  
0000000000i[ ] installing win32 module  
0000000000i[ ] using log file bochsout.  
  
Initialize Device Manager...OK.  
Initialize File System...OK.  
Initialize File Manager...OK.  
Unix V6++ FileSystem Loaded.....OK  
Process 1 exccing  
  
CTRL + 3rd button enables mouse  IPS: 14.400M  NUM: CAPS  SCRL IDG-N
```

3. 进入 bin 文件夹，通过“ls”指令打印出支持的 UNIX 指令如下图：

Bochs for Windows - Display

Directory '/':

dev	Shell.exe	bin	demos	etc	usr	var	
cat	cat.exe	cat1.exe	cp	cp.exe	cpfile.exe	date	date.exe
echo	echo.exe		forks.exe	ls	ls.exe	malloc.exe	
mkdir	mkdir.exe	newsig.exe	perf	perf.exe	rm	rm.exe	
shutdown	shutdown.exe	sig.exe	sigTest.exe	stack.exe	test.exe		

Directory '/bin':

cat cat.exe cat1.exe cp cp.exe cpfile.exe date date.exe

echo echo.exe forks.exe ls ls.exe malloc.exe

mkdir mkdir.exe newsig.exe perf perf.exe rm rm.exe

shutdown shutdown.exe sig.exe sigTest.exe stack.exe test.exe

trace trace.exe

[/bin#date

14-Sep-2023 20:12:7 (THUR)

[/bin#

end sleep

Process 4 (Status:5) end wait

CTRL + 3rd button enables mouse IPS: 14.400M NUM CAPS SCRL ID:0-N

- #### 4. 通过“date”指令查看日期:

Bochs for Windows - Display

Directory '/' :

dev	Shell.exe	bin	demos	etc	usr	var	
[/] # ls							
[/ bin] # ls							
Directory '/' / bin' :							
cat	cat.exe	cat1.exe	cp	cp.exe	cpfile.exe	date	date.exe
	echo	echo.exe	forks.exe	ls	ls.exe	malloc.exe	
mkdir	mkdir.exe	newsig.exe	perf	perf.exe	rm	rm.exe	
shutdown		shutdown.exe	sig.exe	sigTest.exe	stack.exe	test.exe	
	trace	trace.exe					

[/ bin] # _

end sleep

Process 2 (Status:5) end wait

Process 1 finding dead son. They are Process 3 (Status:3) wait until child proc

ess Exit! Process 3 execing

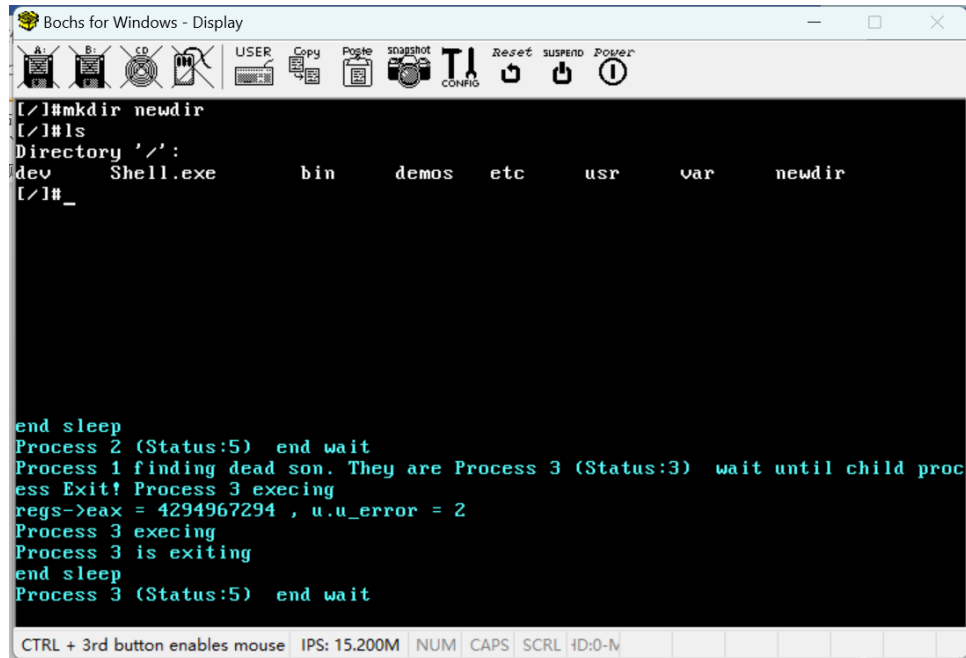
Process 3 is exiting

end sleep

Process 3 (Status:5) end wait

CTRL + 3rd button enables mouse IPS: 12.000M NUM CAPS SCRL 4D:0-IV

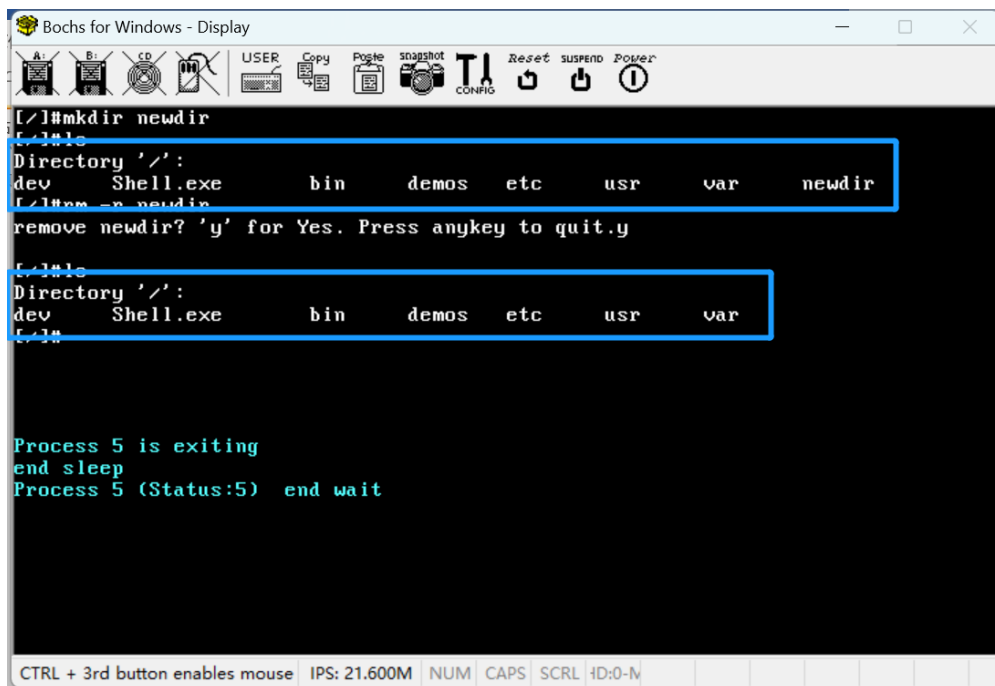
5. 通过“mkdir”指令创建新文件夹“dir”:



```
[/]#mkdir newdir
[/]#ls
Directory '/':
dev      Shell.exe      bin      demos    etc      usr      var      newdir
[/]#

end sleep
Process 2 (Status:5) end wait
Process 1 finding dead son. They are Process 3 (Status:3) wait until child proc
ess Exit! Process 3 execing
regs->eax = 4294967294 , u.u_error = 2
Process 3 execing
Process 3 is exiting
end sleep
Process 3 (Status:5) end wait
```

6. 通过“rm”指令删除文件夹:

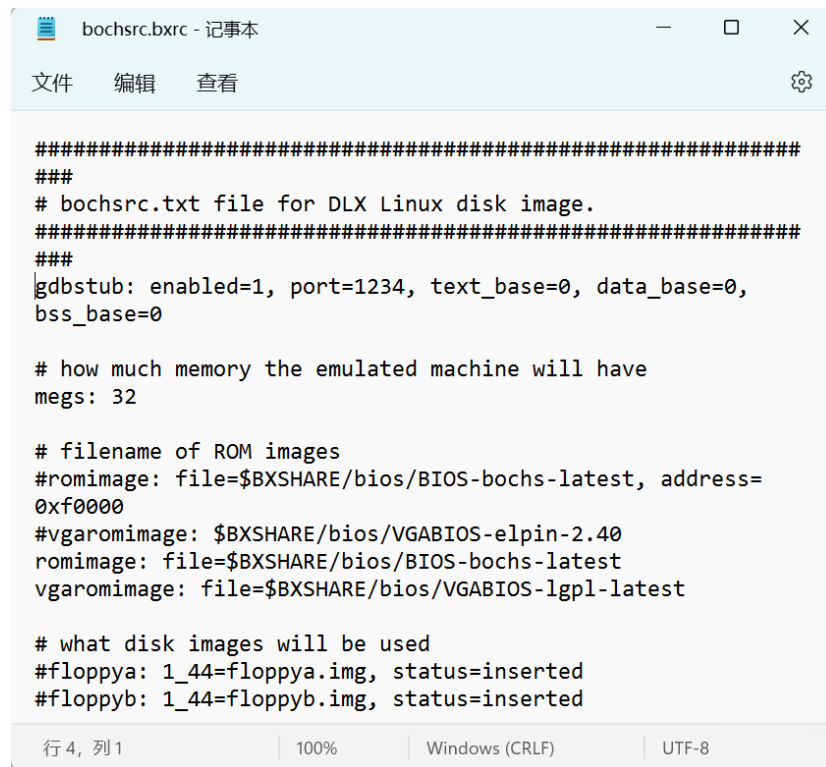


```
[/]#mkdir newdir
[/]#ls
Directory '/':
dev      Shell.exe      bin      demos    etc      usr      var      newdir
[/]#rm -r newdir
remove newdir? 'y' for Yes. Press anykey to quit.y
[/]#ls
Directory '/':
dev      Shell.exe      bin      demos    etc      usr      var
[/]#

Process 5 is exiting
end sleep
Process 5 (Status:5) end wait
```

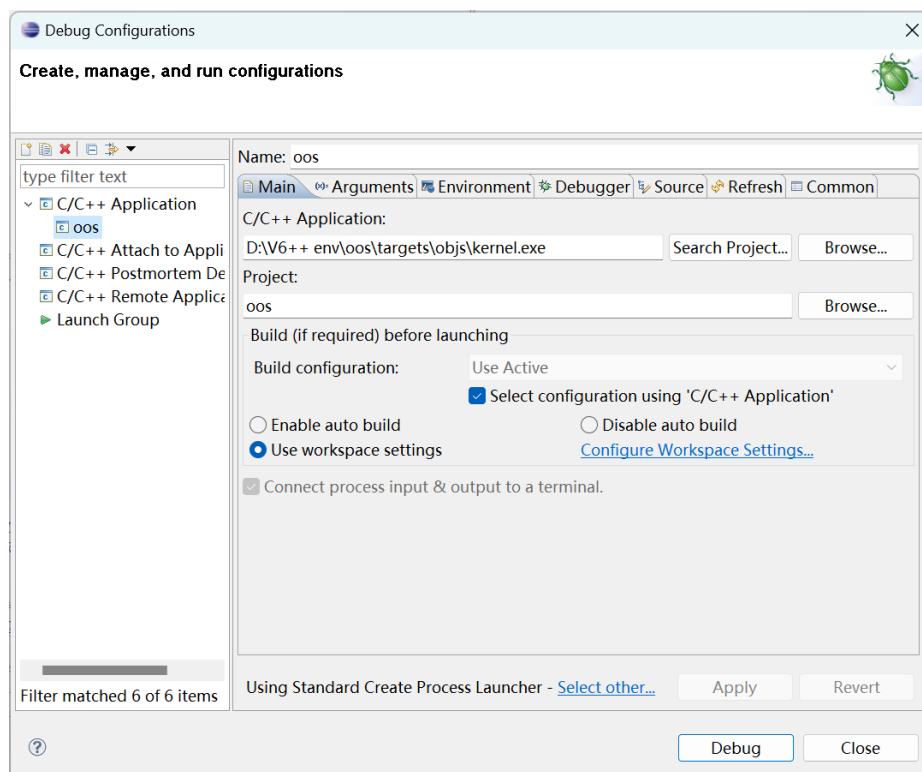
二、 依照实验指导过程，完成 UNIX V6++的 eclipse 远程调试环境的安装与配置，关键步骤截图说明。

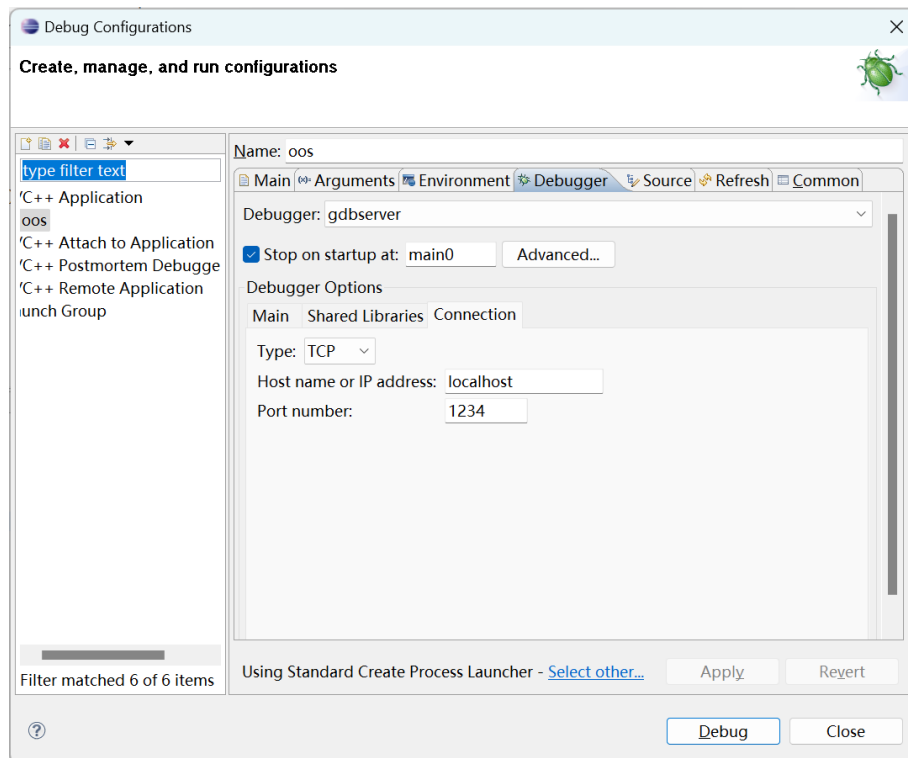
1. 开启调试选项：



```
#####  
###  
# bochsrc.txt file for DLX Linux disk image.  
#####  
###  
gdbstub: enabled=1, port=1234, text_base=0, data_base=0,  
bss_base=0  
  
# how much memory the emulated machine will have  
megs: 32  
  
# filename of ROM images  
#romimage: file=$BXSHARE/bios/BIOS-bochs-latest, address=  
0xf0000  
#vgaromimage: $BXSHARE/bios/VGABIOS-elpin-2.40  
romimage: file=$BXSHARE/bios/BIOS-bochs-latest  
vgaromimage: file=$BXSHARE/bios/VGABIOS-lgpl-latest  
  
# what disk images will be used  
#floppya: 1_44=floppya.img, status=inserted  
#floppyb: 1_44=floppyb.img, status=inserted  
  
行 4, 列 1    100%    Windows (CRLF)    UTF-8
```

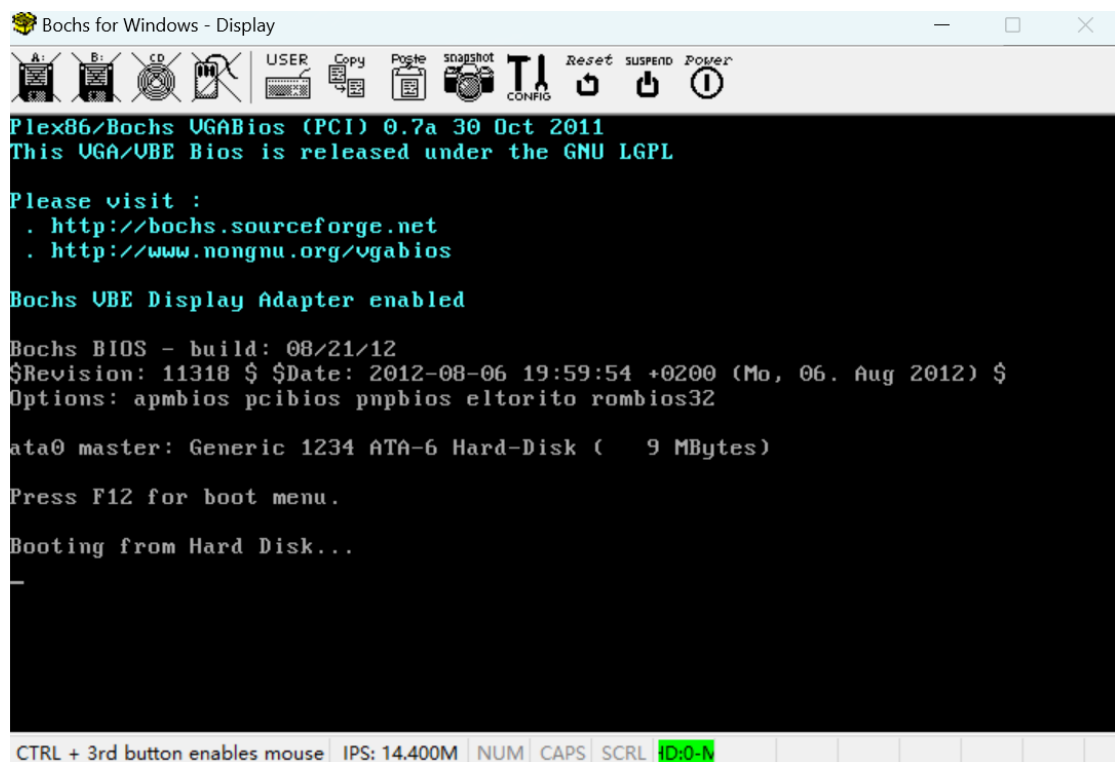
2. 配置 gdb 远程调试：



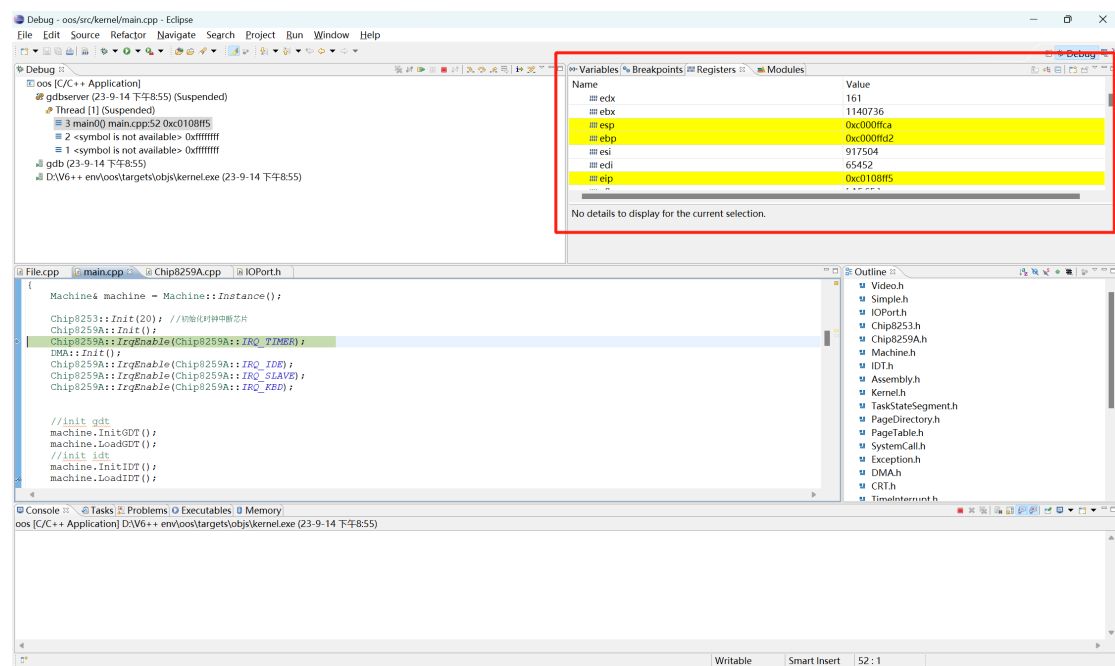


三、 通过自己设置断点，观察 UNIX V6++的调试运行，查看变量与寄存器的值，截图说明。

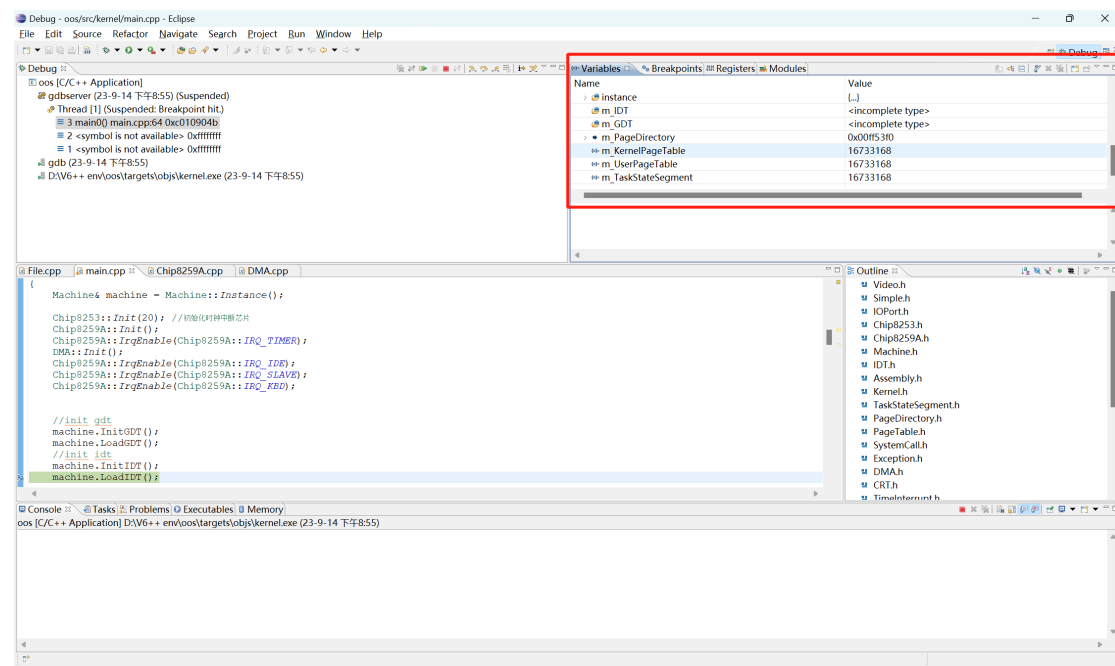
1. 开始调试：



2. 通过调试工具查看寄存器的值:



3. 通过调试工具查看变量的值:



四、 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。

boot	包含 <u>引导操作系统</u> 的相关文件。这些文件负责启动计算机并加载操作系统内核。
dev	包含与 <u>设备驱动程序</u> 相关的文件。用于与计算机硬件交互，控制和管理各种设备。
fs	包含 <u>文件系统</u> 相关的代码和文件。文件系统负责管理磁盘上的文件和目录结构，以及文件的读取和写入操作。
include	包含操作系统项目中使用的 <u>头文件</u> 。头文件包含函数和数据结构的声明，以供其他代码文件引用和使用。
interrupt	包含与 <u>中断处理</u> 相关的代码和文件。用于处理各种中断以确保操作系统稳定可靠。
kernel	包含操作系统 <u>内核</u> 的主要代码文件。内核是操作系统的核心部分，负责管理系统资源、进程调度、内存管理等关键任务。
lib	包含操作系统项目所需的 <u>库文件</u> 。库文件包含一些常用的函数和工具，以便在操作系统的不同部分共享和重复使用。
machine	包含与 <u>硬件架构</u> 相关的代码文件。不同的硬件架构需要不同的代码来管理硬件资源，用于处理特定硬件平台的问题。
mm	包含 <u>内存管理</u> 相关的代码和文件。内存管理是操作系统的重要组成部分，该部分负责分配、释放和管理系统内存。
pe	包含与 <u>解析和加载可执行文件</u> （PE 文件）相关的代码和文件。用于解析 PE 文件头，加载代码段和数据段，管理内存分配等

	操作，以便操作系统能够正确执行和管理这些可执行文件。
proc	包含与 <u>进程管理</u> 相关的代码和文件。进程管理涉及到进程的状态跟踪、进程通信等操作。
program	包含一些 <u>系统程序</u> （如预置的 UNIX 指令）或示例程序的源代码。这些程序可以用于测试和演示操作系统的功能。
shell	包含操作系统的 <u>命令行 shell</u> 的代码和文件。命令行 shell 允许用户与操作系统进行交互，执行命令和管理文件。
test	包含用于 <u>测试</u> 操作系统各个部分的测试代码和脚本。
tty	包含与 <u>终端设备</u> （TTY，即文本终端）相关的代码和文件。终端是用户与操作系统进行交互的界面。