



# Linux0.11 可视化设计



## 数据的提取

- 基本思路：调试。

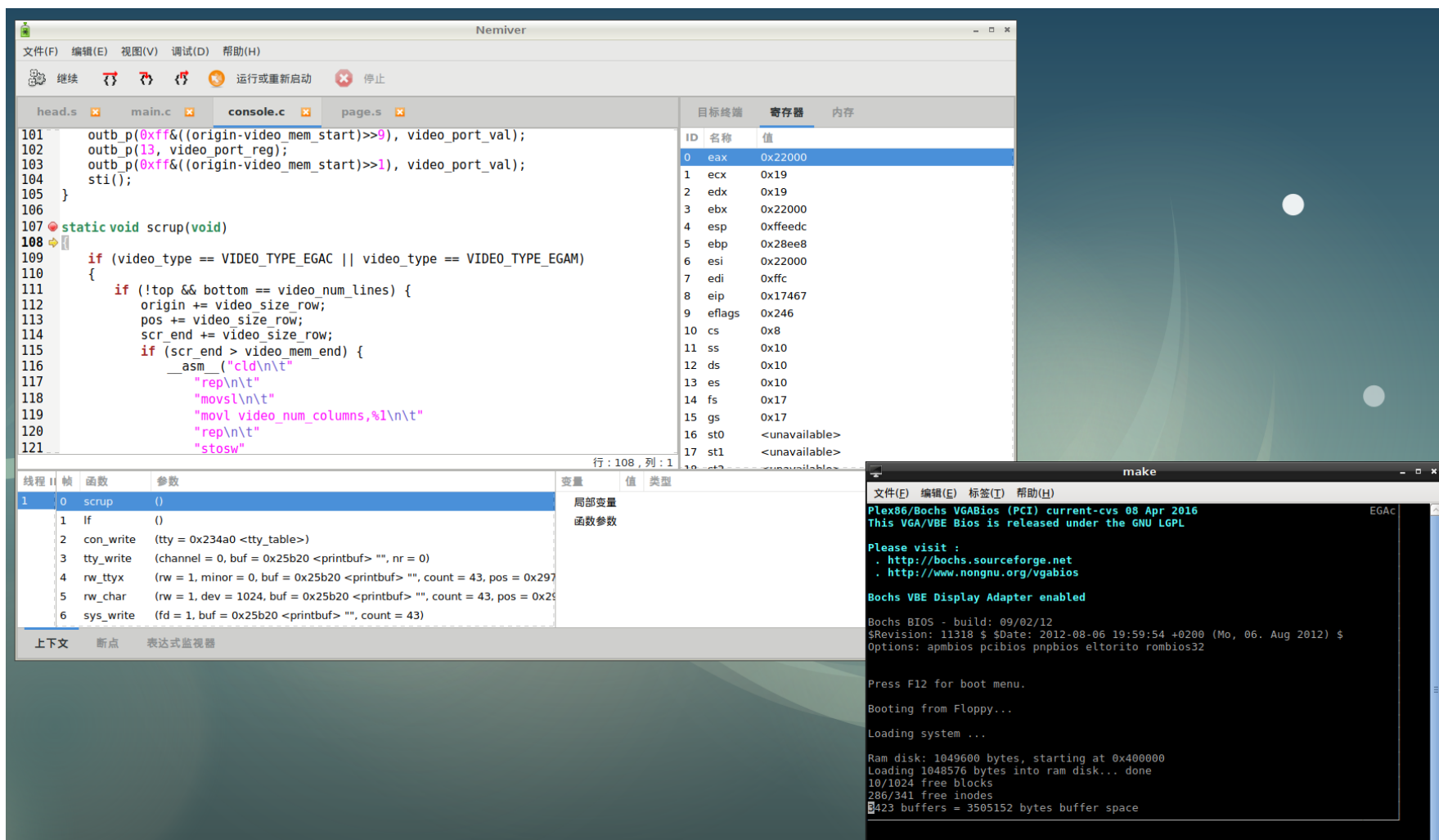
# 使用 **gcc** 调试 **Linux0.11** 的代码

- 机器中运行的 linux0.11 是不包含调试信息的，无法直接调试。
- ld 在链接时可以将调试信息单独输出到其他文件。
- 使用 gdb 加载这些调试信息。

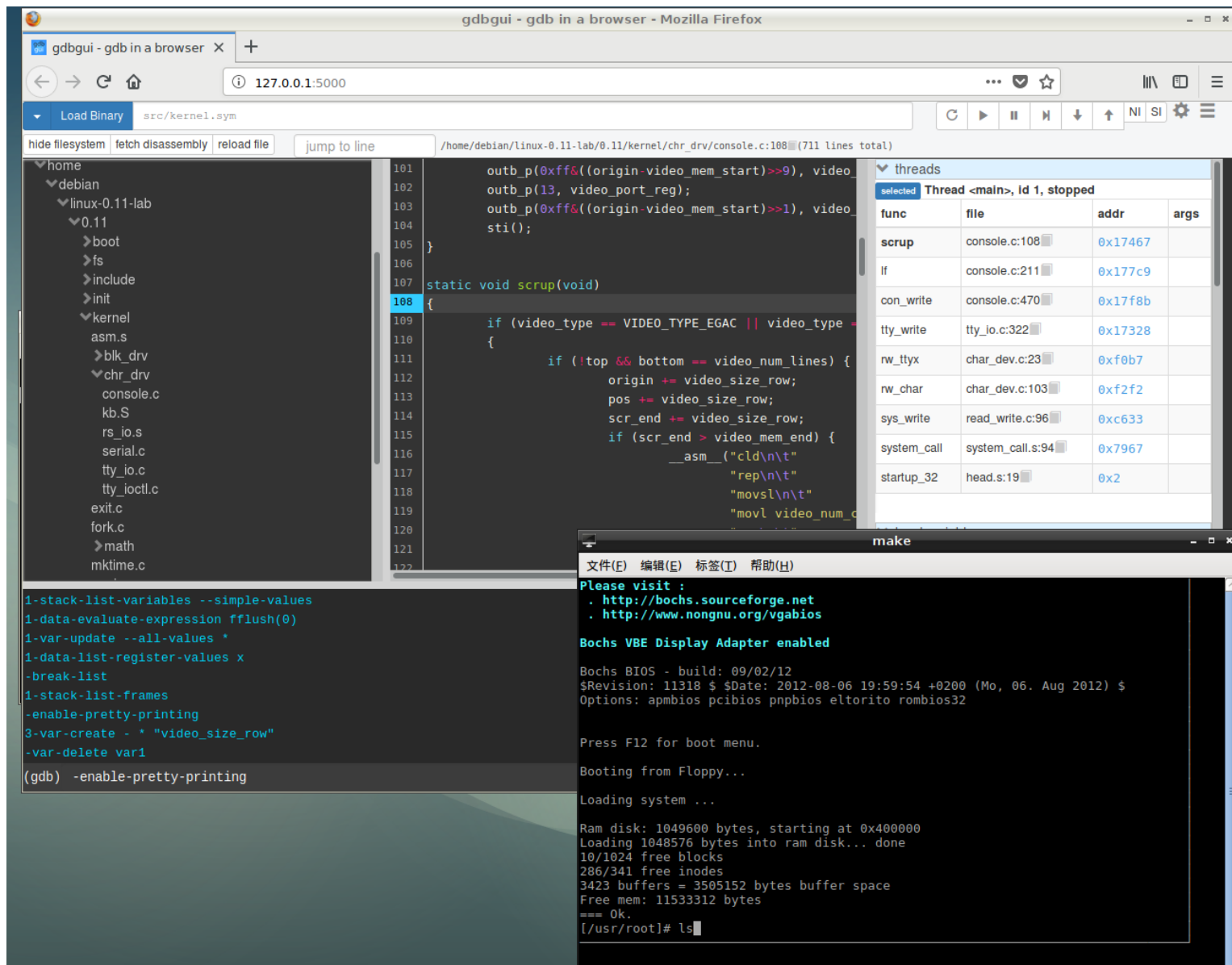
## 图形化的单步调试

- 目的：预先了解涉及的操作中执行了什么代码，确定有价值观察的部分。
- 理论上使用任何支持 gdb 的图形化调试工具都可以。
- 根据功能特性，最终确定选择。

# 截图



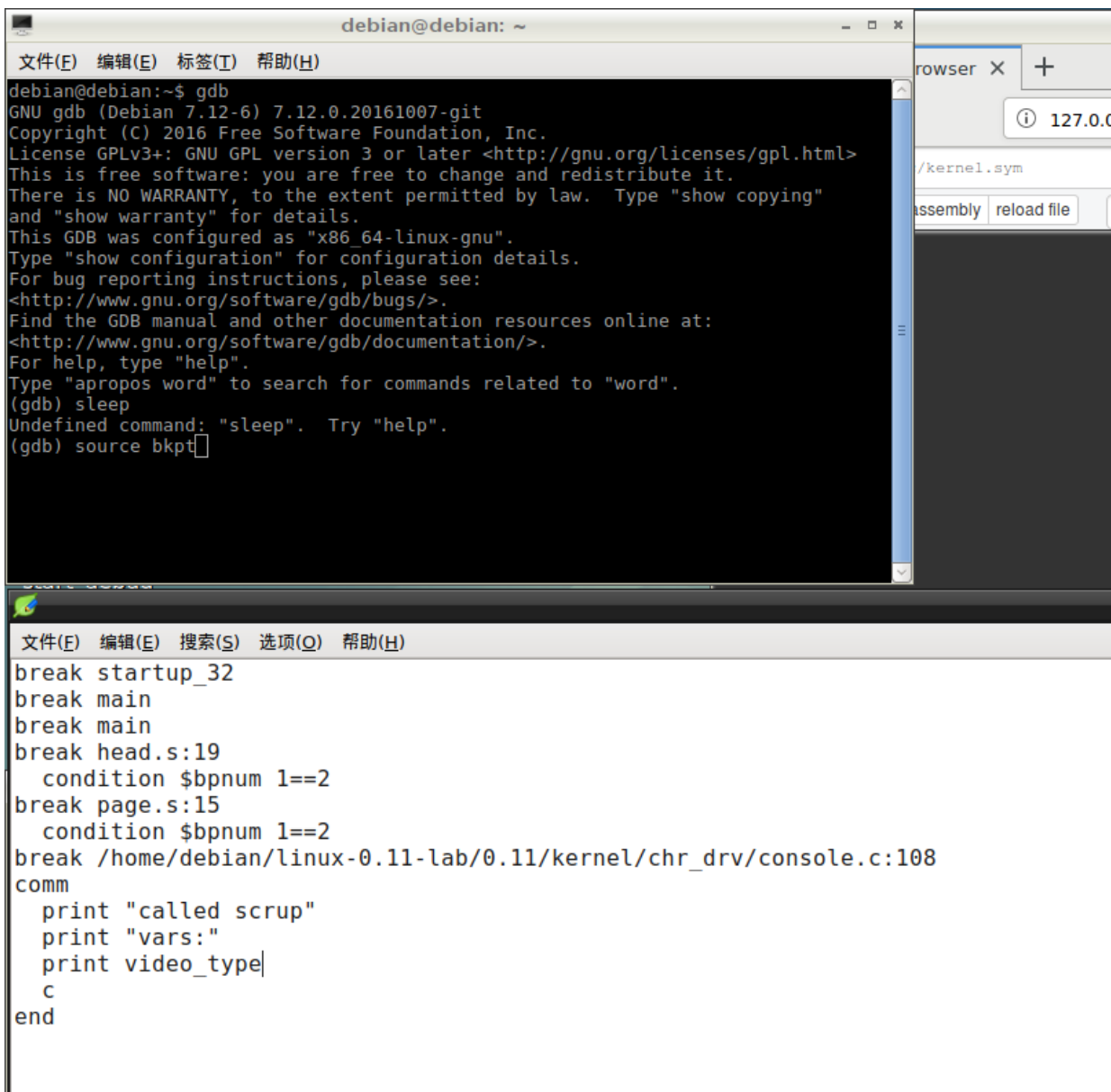
# 截图 2



## 输出

- gdb 支持脚本。
- 使用 gdb 调用脚本，重定向 gdb 的输出到文件。
- 后期处理输出，使得输出更加容易被处理。

# 截图 3



The screenshot shows a terminal window with a menu bar (文件(E), 编辑(E), 标签(T), 帮助(H)) and a title bar (debian@debian: ~). The terminal output shows the GDB startup sequence, including the version (7.12.0.20161007-git), copyright information, and license details. The user enters the command 'sleep', which results in an 'Undefined command: "sleep". Try "help".' error. The user then enters 'source bkpt', which loads a list of breakpoints from a file. The breakpoints are as follows:

```
break startup_32
break main
break main
break head.s:19
    condition $bpnum 1==2
break page.s:15
    condition $bpnum 1==2
break /home/debian/linux-0.11-lab/0.11/kernel/chr_drv/console.c:108
comm
    print "called scrup"
    print "vars:"
    print video_type|
    c
end
```

Below the terminal window, there is a window titled 'State view' with a menu bar (文件(E), 编辑(E), 搜索(S), 选项(O), 帮助(H)). It shows the same list of breakpoints as the terminal window.



## 数据提取方案的优缺点分析

- 优点：完全没有修改 linux0.11 内部代码，保证了 linux0.11 的纯正和完整。
- 缺点：linux 在运行过程中偶尔改变内存布局，使得某些调试信息失效，可以预先生成 linux 运行过程中不同阶段的调试信息，使用 gdb 动态加载调试信息。
- 调试过程中有部分代码的运行过程令人疑惑，可能存在其他问题，这些问题都需要解决。

## 可视化

- 以树的形式表示源码文件或代码块。
- 当前正在执行的部分展开并用特殊颜色表示，其他部分折叠。
- 在发生调用时使用箭头和箭头上的文字表示调用的目标以及传递的参数等。
- 在四周显示一些其他的有价值的信息，比如终端输出、调用统计等。
- 以上为初步设想，若受到其他同学的启发有可能会进行一定的修改。