https://www.cnblogs.com/calvinK/p/9263696.html

## **centos7下安装lldb，dotnet netcore 进程生成转储文件，并使用lldb进行分析**

随着netcore应用在linux上部署的应用越来越多，碰到cpu 100%，内存暴涨的情况也一直偶有发生，在windows平台下进程管理器右键转储，下载到本地使用windbg或者直接vs分析都比较方便。而在linux平台下因为一直接触的不深，所以对这一块也一直没有比较好的了解。所以接下来的文章将对在centos7下安装lldb，生成转储以及调试分析进行一些简单说明。  
还有就是一般产线的机器也不太会有可以直接调试的机会，所以真出现问题也只能在产线机器dump进程，然后下载到本地来慢慢分析。  
PS: 请参考官方爸爸的官方安装脚本:[https://github.com/dotnet/diagnostics/tree/master/documentation/lldb](https://github.com/dotnet/diagnostics/tree/master/documentation/lldb" \t "https://www.cnblogs.com/calvinK/p/_blank)

环境说明：  
os：centos7  
dotnet :2.1.1。查看官方文档2.0.0只能使用lldb 3.6；2.1以上必须是3.9.0；所以特别要注意版本问题，一个是createdump 2.0的有bug会失败。二个是dotnet版本和lldb版本要匹配  
被调试分析的应用也是用2.1跑起来的。

### **测试目标程序**

yum install dotnet-sdk-2.1

dotnet new mvc

vi /mvc.csproj#netcoreapp2.0 to netcoreapp2.1#PackageReference Include="Microsoft.AspNetCore.All" Version="2.0.0" to Version="2.1.1"

dotnet restore

dotnet build

dotnet ./bin/Debug/netcoreapp2.1/mvc.dll

### **centos7 升级GCC，安装cmake**

[centos7 升级GCC版本到7.3.0](https://www.cnblogs.com/calvinK/p/9263297.html" \t "https://www.cnblogs.com/calvinK/p/_blank)  
[centos7 安装cmake](https://www.cnblogs.com/calvinK/p/9263320.html" \t "https://www.cnblogs.com/calvinK/p/_blank)

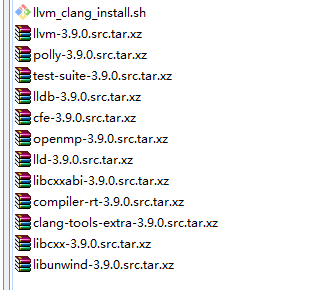
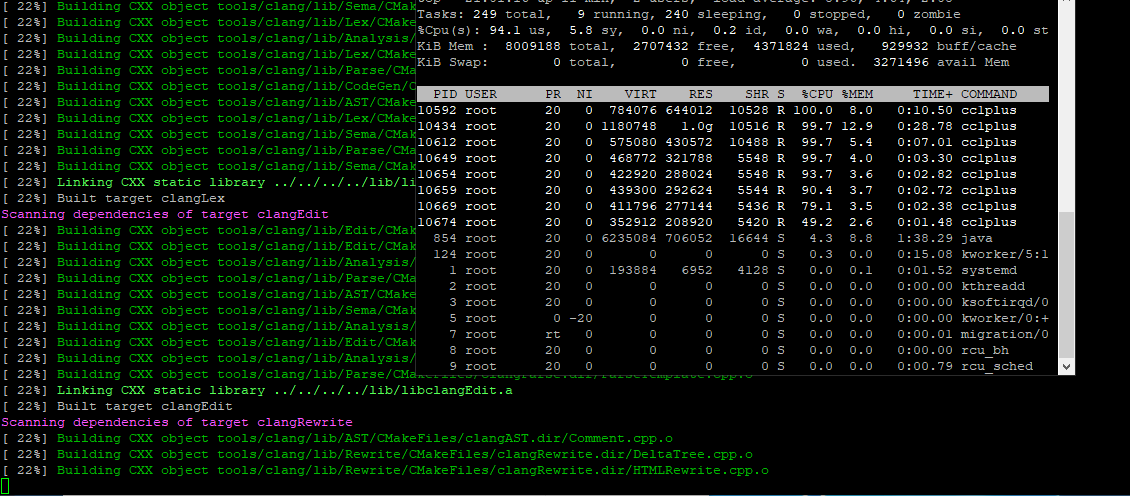
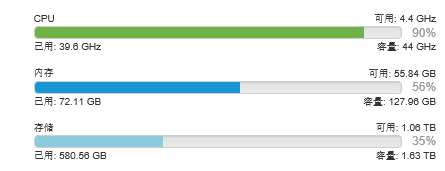
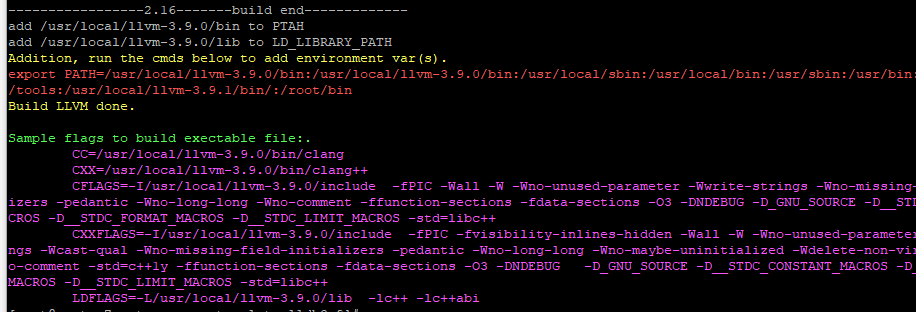
### **centos7下安装lldb调试工具**

最开始直接使用给力网友的脚本进行安装(脚本地址查看文章结尾参考资料)，后发现3.9.1不能调试分析netcore应用，必须要3.9.0，所以在给力网友的脚本上略作修改后使用。修改后脚本地址[https://github.com/czd890/shell/blob/master/llvm\_clang\_lldb/3.9.0/llvm\_clang\_install.sh](https://github.com/czd890/shell/blob/master/llvm_clang_lldb/3.9.0/llvm_clang_install.sh" \t "https://www.cnblogs.com/calvinK/p/_blank)。主要修改几个地方：把lldb，libunwind移动到build\_llvm\_toolchain中，一次性安装。check\_and\_download方法中检查本地是否已下载源码包的检查略作修改，只判断指定版本，编译的时候修改为make -j8（我本地机器8核)。

脚本大概思路就是下载如下所表示的组件所有源码，除llvm外的其他组件源代码解压到llvm/tools目录下,这样子源代码就全部准备好  
BUILD\_TARGET\_COMPOMENTS="llvm clang compiler\_rt libcxx libcxxabi clang\_tools\_extra lldb lld libunwind";  
接下来就是编译的过程了。

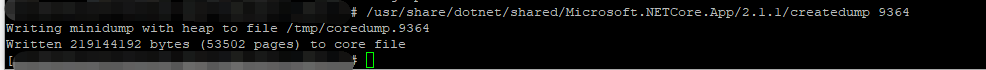
#安装一些必要的依赖组件

yum install libedit-devel libxml2-devel ncurses-devel python-devel swig#执行根据给力网友的脚本修改后的脚本

当然如果脚本下载速度慢，也是可以自己下载后上传的目录的。具体下载地址查看文章尾部参考资料 llvm，clang，lldb源代码下载地址(3.9.0)  
准备源代码差不多就如下图。然后 sh llvm\_clang\_install.sh开始执行脚本；  
默认安装目录在 PREFIX\_DIR=/usr/local/llvm-$LLVM\_VERSION;。也就是是 /usr/local/llvm-3.9.0;可以在脚本的最开始对此进行修改。  
  
开始执行，又是一段漫长的等待时间，8核并发编译，耗费了估计得有1-2个小时。  
  
刀片机的CPU都跑满了！！！  
  
出去吃完饭后回来，就看到完成拉。具体的path路径可以选择加不加都可以，加的话，直接/etc/profile export PATH=$PATH:llvm-path/bin即可  
  
lldb安装完成，我们的工作就完成一大半拉。

### **dotnet netcore应用如何生成内存转储文件**

/usr/share/dotnet/shared/Microsoft.NETCore.App/2.1.1/createdump 9364

  
具体命令解释  
createdump [options] pid  
-f, --name - dump path and file name. The pid can be placed in the name with %d. The default is "/tmp/coredump.%d"  
-n, --normal - create minidump (default).  
-h, --withheap - create minidump with heap.  
-t, --triage - create triage minidump.  
-u, --full - create full core dump.  
-d, --diag - enable diagnostic messages.

### **使用lldb调试分析netcore应用内存转储文件**

#官方文档上是这样写的。

/usr/local/llvm-3.9.0/bin/lldb -O "settings set target.exec-search-paths /usr/share/dotnet/shared/Microsoft.NETCore.App/2.1.1" \

-o "plugin load /usr/share/dotnet/shared/Microsoft.NETCore.App/2.1.1/libsosplugin.so" \

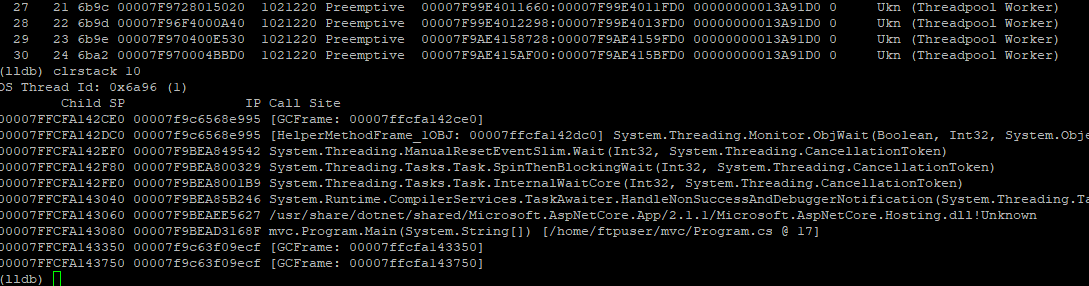
--core /opt/dump\\_file/mvcdumpmindump /usr/share/dotnet/dotnet

#网友调试参考博客上是这样写的。

/usr/local/llvm-3.9.0/bin/lldb dotnet \

-c /opt/dump\\_file/mvcdumpmindump \

-o "plugin load /usr/share/dotnet/shared/Microsoft.NETCore.App/2.1.1/libsosplugin.so"

2种写法都是可行的。然后具体的调试分析指令什么的都在[coreclr调试说明指导文档](https://github.com/dotnet/coreclr/blob/master/Documentation/building/debugging-instructions.md" \l "sos-commands" \t "https://www.cnblogs.com/calvinK/p/_blank)有说明。  


参考资料：  
coreclr调试说明指导文档  
[https://github.com/dotnet/coreclr/blob/master/Documentation/building/debugging-instructions.md](https://github.com/dotnet/coreclr/blob/master/Documentation/building/debugging-instructions.md" \t "https://www.cnblogs.com/calvinK/p/_blank)  
coreclr生成dmp说明指导文档  
[https://github.com/dotnet/coreclr/blob/master/Documentation/botr/xplat-minidump-generation.md](https://github.com/dotnet/coreclr/blob/master/Documentation/botr/xplat-minidump-generation.md" \t "https://www.cnblogs.com/calvinK/p/_blank)  
llvm，clang，lldb源代码下载地址(3.9.0)  
[http://releases.llvm.org/download.html#3.9.0](http://releases.llvm.org/download.html" \l "3.9.0" \t "https://www.cnblogs.com/calvinK/p/_blank)  
lldb源码安装指导文档  
[http://lldb.llvm.org/build.html#BuildingLldbOnLinux](http://lldb.llvm.org/build.html" \l "BuildingLldbOnLinux" \t "https://www.cnblogs.com/calvinK/p/_blank)  
llvm源码安装指导文档  
[http://releases.llvm.org/3.9.0/docs/GettingStarted.html](http://releases.llvm.org/3.9.0/docs/GettingStarted.html" \t "https://www.cnblogs.com/calvinK/p/_blank)  
网友centos7安装llvm，clang，lldb等给力脚本  
[https://github.com/owent-utils/bash-shell/blob/master/LLVM%26Clang%20Installer/3.9/installer.sh](https://github.com/owent-utils/bash-shell/blob/master/LLVM&Clang Installer/3.9/installer.sh" \t "https://www.cnblogs.com/calvinK/p/_blank)  
网友调试参考博客文章  
[使用SOS调试工具检查应用程序状态](https://faithlife.codes/blog/2018/01/using-sos/" \t "https://www.cnblogs.com/calvinK/p/_blank)