Debugging C/C++ programs

SOmetimes a program dies, and leaves a file of its last memory state. Using the gdb debugger we can see what happened

recently I installed firefox 3 from ports. Then it died when I tried printing.

```
$gdb core firefox-bin.core
```

```
# gdb core firefox-bin.core
GNU gdb 6.1.1 [FreeBSD]
Copyright 2004 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public Li-
cense, and you are
welcome to change it and/or distribute copies of it under certain condi-
tions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB.  Type "show warranty" for details.
This GDB was configured as "i386-marcel-
freebsd"...core: No such file or directory.

Core was generated by ‘firefox-bin’.
Program terminated with signal 11, Segmentation fault.
#0  0x299c5c0b in ?? ()
```

The problem here is the last call made before everything went blooey was in memory address 0x299c5c0b (Hex for 698113035). unfortunately if this binary had been compiled with debugging symbols I could see the name of that function not ??

Normally bt (backtrace) would help but it is useless too...

so I recompiled firefox with debugging symbols, and tried to break it again. (I altered the options file in /var/db/ports/firefox3/ - WITHOUT_DEBUG=false. Yes double megatives do not help my mental state right now)

bibliography

http://www.freebsd.org/doc/en/books/developers-handbook/debugging.html http://www.freebsd.org/doc/en/books/de
handbook/kerneldebug-gdb.html http://www.unix.com/unix-advanced-expert-users/19128-how-do-core-
dump-analysis.html http://www.unknownroad.com/rtfm/gdbtut/gdbuse.html