

Segmentation Fault

19.05.2024 04:12:11

FAQ-Artikel-Ausdruck

Kategorie:	RRZE: HPC	Bewertungen:	0
Status:	öffentlich (Alle)	Ergebnis:	0.00 %
Sprache:	de	Letzte Aktualisierung:	17:31:23 - 19.02.2009

Symptom (öffentlich)

Problem (öffentlich)

Mein Programm ist mit einem "Segmentation Fault" abgestürzt, aber ich weiß nicht genau, wo.

Lösung (öffentlich)

Abgesehen vom immer möglichen "printf-Debugging" gibt es noch die schnelle "post-mortem-Analyse" mittels Core Dump. Normalerweise wird nach einem Segfault ein sog. Corefile geschrieben (Filename "core" oder "core.XXXX" mit XXXX=PID des Prozesses). Dieses Corefile kann man zusammen mit dem Binary einem Debugger geben, der dann in der Lage ist, mindestens die Funktion zu lokalisieren, in der der Absturz passierte (wenn das Programm mit Debugging-Option übersetzt wurde, ist auch die Codezeile verfügbar). Mit dem GNU-Debugger sieht das so aus:

```
$ gdb a.out core
GNU gdb 5.2.1
Copyright 2002 Free Software Foundation, Inc.
[...]
Core was generated by `./a.out'.
Program terminated with signal 11, Segmentation fault.
[...]
#0 0x0804856b in intersect (x=2.1000000000000001, y=1.3,
z=1.7, dx=-0.58851242127038994, dy=-0.78466509831745435,
dz=-0.19481738498635751, maxp=0xbffff2a0) at raytrace.c:29
29 xx = *o++ - x; yy = *o++ - y; zz = *o++ - z;
(gdb) where
#0 0x0804856b in intersect (x=2.1000000000000001, y=1.3, z=1.7,
dx=-0.58851242127038994, dy=-0.78466509831745435,
dz=-0.19481738498635751, maxp=0xbffff2a0) at raytrace.c:29
#1 0x080486f3 in shade (x=2.1000000000000001, y=1.3, z=1.7,
dx=-0.58851242127038994, dy=-0.78466509831745435, dz=-0.19481738498635751,
de=0) at raytrace.c:46
#2 0x08048b2d in calc_tile (size=1000, xstart=0, ystart=0, tilesize=200,
tile=0x80492f0 "") at raytrace.c:90
#3 0x08048cc0 in main () at raytrace.c:124
#4 0x4005a4f2 in __libc_start_main () from /lib/libc.so.6
(gdb) quit
```

Dieses Programm war mit -g übersetzt worden, deshalb wird gleich die Codezeile (29) angezeigt, in der der Segfault passierte. Das Kommando "where" liefert dann noch den aktuellen Call-Stack mit den Namen und Argumenten aller bis zum Stop gerufenen Funktionen.