

# **Advanced Programming in the UNIX Environment**

## **Week 05, Segment 9: Unix Development Tools: Using gdb(1)**

**Department of Computer Science  
Stevens Institute of Technology**

**Jan Schaumann**

jschauma@stevens.edu

<https://stevens.netmeister.org/631/>

```
a  
atime (jschauma)  
longago (jschauma)  
inthefuture (jschauma)  
subdir (jschauma)  
executable (jschauma)  
suid:sgid (jschauma)  
missing-exec (jschauma)  
  (jschauma)  
    (jschauma)  
bar (jschauma)  
wd0 (root)  
noowner (1234)  
nogroup (jschauma)  
ls-LR (root)  
null (root)  
hole.c (jschauma)  
a.out (jschauma)  
file.hole (jschauma)  
a.out.core (jschauma)  
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa  
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa  
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa  
aaaaaaaaaaaaaaaa (jschauma)  
apue$
```

## Using a debugger

---

The purpose of a debugger such as `gdb(1)` is to allow you to see what is going on “inside” a program while it executes or what it was doing at the moment it crashed.

- we need to enable debugging symbols via the “-g” flag to the compiler
- merely running the failing program inside the debugger lets you pinpoint the exact place in the code (function name and arguments, file name, line number) where your program crashed
- we can inspect variables and even call functions