

# Felsökning

Niclas Börlin

[niclas.borlin@cs.umu.se](mailto:niclas.borlin@cs.umu.se)

Datavetenskap, Umeå Universitet

2022-01-27 Tor

# Olika typer av fel i program

- ▶ Syntaxfel
  - ▶ Upptäcks av kompilator/interpretator
- ▶ Programkörningsfel (*run-time error*)
  - ▶ Koden är syntaktiskt korrekt men programmet kraschar när man kör det.
- ▶ Logiska fel
  - ▶ Koden är syntaktiskt korrekt och programmet kör som det ska men resultatet blir inte det man tänkt sig.

# Hur hittar man orsakerna till fel?

- ▶ Läsa meddelanden från kompilator/interpretator, koden och kommentarerna en extra gång
- ▶ Läsa **första** felmeddelandet igen
- ▶ Lägga till spårutskrifter i programmet.
- ▶ **Debugger**

# Debugger

- ▶ En debugger gör det möjligt att
  - ▶ **Stega igenom** koden rad för rad (*single-step*).
  - ▶ Följa variablers **värden** och hur de **förändras** under körning (*watch*).
  - ▶ **Förändra** variablers värden under körning.
  - ▶ Sätta **brytpunkter** (*breakpoints*) som bestämmer var exekveringen ska stanna.
- ▶ Under stegningen kan man välja mellan att
  - ▶ **gå in** i subrutiner/procedurer (*step in*),
  - ▶ att **gå förbi** subrutinerna (*step over*),
  - ▶ **hoppa ut** ur en subrutin (*step out*).

```

1 #include <stdio.h>
2
3 int main(void){
4     int fak;
5     int result = 1;
6
7     printf("This program computes n-factorial.\nPlease specify n: ");
8     scanf("%d", &fak);
9
10    while (fak > 1);
11    {
12        result *= fak;
13        fak++;
14    }
15    printf("Answer: %d\n", result);
16

```

Line: 3, Column: 1

Variable

Value	Type
In scope expressions	
Out of scope expressions	

Target Terminal Context Breakpoints Registers Memory Expression Monitor

Debug	Help
Run or Restart	Shift+F5
Stop	F9
Continue	F5
Run to Cursor	F11
Next	F6
Step	F7
Step Out	Shift+F7
Step Into asm	Ctrl+I
Step Over asm	Ctrl+N
Jump to a Given Location	Ctrl+J
Set Breakpoint with Dialog...	Shift+Ctrl+B
Set Breakpoint...	Ctrl+B
Set Breakpoint	F8
Disable Breakpoint	Shift+F8
Set Countpoint	
Set Watchpoint with Dialog...	Ctrl+T
Inspect an Expression	F12
Call a Function	Ctrl+E
Show Global Variables	Ctrl+G
Refresh Locals	
Show Assembly	
Switch to Assembly	Ctrl+A
Switch to Source	Shift+Ctrl+A

# Tips för debugging

- ▶ Gör lite detektivarbete för att försöka att ringa in felet
  - ▶ Testa olika argument till programmet. Är det några som fungerar?
  - ▶ Händer ingenting? Låter som en oändlig loop
  - ▶ Kraschar det med segmentation fault? Låter som pekar-fel (här kan valgrind hjälpa ibland)
- ▶ Sätt breakpoints där du misstänker att nånting går fel
- ▶ Försök inte gissa felet slumpmässigt, då slösar du bara tid
- ▶ Tänk till när du fixar en bugg. Har du löst hela problemet eller måste du göra nåt mer?
  - ▶ Undvik att göra massa små inkrementella ändringar för att se när felet försvinner
- ▶ Kontrollera så att du inte skapat en ny bugg
- ▶ Har du jagat samma bug länge? Trött? Kanske bra att ta en paus?

## Syftet med workshop 3

- ▶ Ni ska känna er bekväma med att hantera en grafisk debugger för att hitta fel i koden
- ▶ Öva lite mer på valgrind för att hitta minnesläckor i koden
- ▶ Förhoppningen är att det ska hjälpa er på labbarna framöver