

Debugging

Vad är

ABC/LP

- Våra program gör oftast inte det vi vill att de skall göra.
- Diverse logiska fel:
 - Iterationer som utförs för få eller för många gånger
 - Variabler som inte har initierats
 - Villkor som inte fungerar
 - Mm.
 - Mm.

- Printf är vår vän när det gäller att få saker att funka fort.
- Ta för vana att:
 - Skriva ut värden på skärmen, t ex efter det vi läst in ett värde skriv ut det på skärmen för att så det inlästes korrekt.
 - Utskrift i början av en funktion (med alla in parametrar)
 - Utskrift av variabler i en iteration ger bra ledtrådar, t ex indexvariabler, värden som beräknas
- De flesta av alla dessa utskrifter kan tas bort ganska omgående.
- Andra kan vara bra att ha kvar i programmet men omslutas av `#ifdef DEBUG`

När

ABC/LP

- Lägg in utskriftssatserna samtidigt som ni skriver programmet.
- Jobbigt att göra i efterhand, men ibland nödvändigt.

gdb

ABC/LP

- Debuggers är en miljö där vi kan exekvera vårt program på ett kontrollerat sätt.
- Vi kan:
 - exekvera en rad i taget
 - skriva ut variablers värden
 - sätta brytpunkter, dvs. exekvera fram tills en viss rad
 - exekvera ett kraschande program och exakt få reda var det kraschade och sedan kolla variabler värde

Hur göra

ABC/LP

- Kompilera källkoden med flaggan `-g`
`gcc -g crasher.c -o crasher`
- Kompilatorn genererar info som debuggern sedan använder.
- Starta gdb
`gdb crasher`
- Nu kan vi t ex sätta en brytpunkt i vårt program
`break 20` (stanna exekveringen på rad 20)
- Starta programmet:
`run`
Breakpoint 1, main () at chrasher.c:20
`20 i = i * 100;`

- Nu kan vi skriva ut värdet som i har med print .
p i
\$1 345
- Om detta inte ger tillräckliga ledtrådar kan vi fortsätta exekveringen
- c eller continue, fortsätt till nästa brytpunkt
- n eller next, exekvera nästa rad
- s eller step, exekvera nästa rad och gå in i funktioner

Några kommandon

ABC/LP

- l eller list
 - p eller print
 - c eller continue
 - s eller step
 - help
 - quit
 - ENTER, kör förra kommandot en gång till.
-
- Kolla den utmärkta manualen hos Gnu:
 - <http://www.gnu.org/software/gdb/documentation/>

- LIVE EXEMPEL, baseras på factorial.c