

Linux som utvecklingsmiljö

Tema 8 Utvecklingsverktyg

HT-13

Petter Lerenius, 790703-0295

1. Makefile justering

För att få med debuginformation i programmet så lade jag till -g flaggan till gcc komandot i makefilen.

2. Debuggning

För att få en första insikt i programmets funktion, så körde jag det utan debugger och såg att utskrifterna inte stämde med vad som kunde förväntas, eftersom additionerna visade fel resultat.

Jag startade sedan debuggern med:

```
$ gdb program
```

Gdb startade då upp och läste in programet program. Första steget i gdb blev att starta programmet med kommandot "Start", varpå gdb startade upp programmet och stannade på första raden i main(), vilket var anropet till test(). För att få en överblick körde jag kommandot "list" som visar programkoden runt omkring där gdb står. Jag insåg då att jag behövde stega mig in i funktionen test med hjälp av kommandot "step".

Väl inne i test(), körde jag list på nytt för att se hur funktionen såg ut, sen tryckte jag på *enter* en extra gång för att se mer. Gdb skriver då ut mer av koden från där den slutade senast. Jag kunde då se var utskrifterna skedde och lade in en brytpunkt (breakpoint) på första utskriftsraden med hjälp av kommandot "break <kodrad>". För att få programmet att fortsätta exekvera fram till brytpunkten använde jag kommandot "continue".

Väl framme vid utskriftsraden använde jag mig av kommandot "print" för att skriva ut värdet av variablerna för att kunna se vad som gått fel. T.ex:

```
print test->buffert1[15]
print test->buffert2[0]
```

Det var då ganska uppenbart för mig vad som var fel i koden och detta kunde rättas till, se nästa punkt. Därefter kunde programmet köras med, ett för mig, förväntat resultat.

För att avsluta gdb körde jag färdigt programmet med "continue" och avslutade gdb med "quit".

3. Rätta koden

Buggarna fanns i filen gdbl.c, vilket framgick av utskrifterna i gdb.

```
test () at lib/gdbl.c:16
```

Det fanns två problem, som jag valde att rätta genom att göra så små ändringar som möjligt i koden.

Det första var att buffert1 var för liten. Både i loopen på rad 20 och i utskriften på rad 28 användes 17:nde elementet i buffert1, (test->buffert1[16]), fast den bara hade 16 element. Så jag utökade storleken med ett på rad 12.

```
int buffert1[16] → int buffert1[17]
```

Det andra problemet var att utskriften på rad 29, skrev ut innehållet i buffert1 istället för buffert2, som såg ut att vara det som förväntade sig enligt tilldelningarna tidigare i koden vilka överrensstämde med utskriften 45+5. Så rad 29 ändrades enligt följande:

```
printf("45 + 5 = %i\n", test->buffert1[0] + test->buffert1[1]);
```

→

```
printf("45 + 5 = %i\n", test->buffert2[0] + test->buffert2[1]);
```