

Screencast

Kommandoradsargument

Sammanfattning

- Ett C-program tar 1 eller flera argument när det körs som fångas via argument till main

```
int main(int argc, char *argv[])
```

- Parametern argc håller antalet argument; "argument counter"
- Parametern argv är en argumentvektor i form av en array av strängar
- Här är ett program som skriver ut sina samtliga kommandoradsargument på terminalen

```
1    #include <stdio.h>
2    int main(int argc, char *argv[]) {
3        do {
4            puts(*argv);
5        } while (*++argv);
6        return 0;
7    }
```

- Studera detta program och förstå hur det fungerar! Hur kan det rimligen terminera?

- **SVAR:** Argumentvektorn `argv` är en array av pekare till strängar (aka en pekare till en pekare eller flera pekare av strängar) vars sista pekare är `NULL`. `NULL` kan omvandlas till heltalsvärdet 0, som alltså är falskt. Varje varv i loopen flyttas pekaren `argv` ”ett pekare åt höger” (adressen ökas med `sizeof(char*)` många bytes), och termineringsvillkoret är att man till sist skall hamna på en pekare vars värde är `NULL`. (Jämför med hur strängar är representerade och hur man terminerar loopar över innehållet i en sträng.)
- Prova att kompilera och köra detta program (det finns i kursens repo). Vad skrivs ut? Notera den första utskriften!
- En implementation som är betydligt tydligare än den föregående använder `argc`:

```
1      #include <stdio.h>
2      int main(int argc, char *argv[]) {
3          for (int index = 0; index < argc; ++index) {
4              puts(argv[index]);
5          }
6          return 0;
7      }
```

Här använder vi en `index`-variabel och förändrar varken `argc` eller `argv` vilket är god sed att undvika.