

# TDIU16: Process- och operativsystemprogrammering

## Felsökning med debugger

Klas Arvidsson, Daniel Thorén, Filip Strömbäck

## 1 Mål

Felsökning är en av de svåraste utmaningarna när man programmerar och att Pintos är ett helt operativsystem med en stor befintlig kodbas gör det inte lättare. Som tur är finns det verktyg som kan hjälpa oss med felsökning. I denna uppgift är det tanken att ni ska få lära er att använda debuggern så att ni lättare kan felsöka er egen kod. Information om hur debuggern fungerar finns i Pintos-Wiki under rubriken "Inledning", se särskilt underrubriken "Felsökning av Pintos".

## 2 Uppgift

Rita och följ exekveringen med alla arrayer och pekare på papper. Vad är det meningen programmet skall göra? Försök använda debuggern för att verifiera sådant du är osäker på. Det är viktigt att förstå allt som händer i minnet. När du hittat felet, rätta det och prova. Det räcker att byta ordning på två rader för att rätta felet. Koden finns även i filen `src/standalone/labx4/debug.c` (använd filen `debug-visualization.c` om du vill köra koden i visualiseringsverktyget).

```
int main()
{
    char str[] = "sihtgubed";
    char *stri = &str[8];
    char *buf[9];
    char **bufi, **bufend;
    bufi = buf;
    bufend = &buf[9];
    while (bufi != bufend) {
        *bufi = stri;
        bufi++;
        stri--;
    }

    while (bufi != buf) {
        *(*bufi) -= 32;
        bufi--;
    }

    while (bufi != bufend) {
        printf("%c", **bufi);
        bufi++;
    }
}
```

För att använda debuggern med Pintos krävs några ytterligare steg för att komma igång. Eftersom Pintos exekverar i en emulator (*qemu*) måste en anslutning upprättas mellan emulatoren och debuggern. Detta återkommer vi till när Pintos är installerat.