

# TDIU16

## Föreläsning 0

Filip Strömbäck, Klas Arvidsson

- 1 Kursinformation
- 2 Introduktion till C

## Resurser

- Kurshemsida: <http://www.ida.liu.se/~TDIU16/>
- Litteratur: *Operating System Concepts* av Silbertschatz, Galvin, Gagne

Examinator	Filip Strömbäck
Assisterer	Klas Arvidsson (även KL) Herman Appelgren (KA) Kasper Landgren
Administratör	Annelie Almquist

# Distansundervisning

Distansundervisning i kursen.

- Föreläsningar - Via Zoom på detta viset
- Laborationer:
  - Handledning: Teams, kösystem
  - Eget arbete: Använd gärna Teams, men välj själva
- Seminarier - Via Teams, som laborationer

## Mål med kursen

Tillämpa teori från TDIU11:

- Hur ser en "vanlig" dator ut?
- Hur körs program?
- Vilka viktiga resurser finns för programkörning?
- Varför finns operativsystem?
- Vad gör operativsystemet? När körs det?

# Mål med kursen

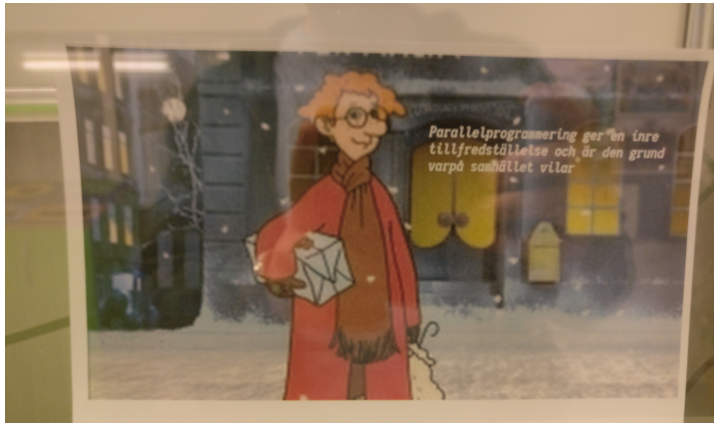
## Nytt innehåll:

- Multitrådning
- Synkronisering
- Programmering i C

## Praktisk erfarenhet:

- Programmera i större system
- Läsa, förstå och modifiera kod som andra har skrivit
- Felsökning

## Mål med kursen



# Examination

DAT1 Skriftlig tentamen med dator, 1hp (U, 3, 4, 5)

LAB1 Laborationer, 3hp (U, G)

- Ett ofärdigt operativsystem (Pintos) skrivet i C
- 80h skall användas
- Arbeta i par som vanligt
- Redovisas muntligt (under labpass)
- Mjuka deadlines ger bonus
- En hård deadline
- Eget arbete utöver schemalagd tid krävs



# Labanmälan

- Arbete sker parvis som vanligt.
- Labanmälan öppen tom. 29/3, anmäl er senast idag!
- Små uppgifts-PDF:er + Wiki

## Tid i kursen

- Kursen är 4hp  $\approx$  100h
- Du kommer att behöva alla 100h. Planera din tid så att du pluggar varje vecka. Lämna inte alla 100h till dagarna före tenta.
- Det finns fristående uppgifter ni kan börja med ifall ni kör fast.

# Planering

Vecka	Fö/Se	Lab
13	Fö: C + Syscall (påsk)	C <sup>1</sup> , halt, exit, console
14	-	console, filhantering
15	Fö: Sema, Se: Process	main, första processen
16	Fö: Lås, cond.	exec, exit
17	Fö: Implementation	exec, exit, wait
18	Fö: Deadlock	Synkronisering
19	-	Synkronisering, accesskontroll
20	Se: Deadlock + tenta	Synkronisering, accesskontroll
21	-	Synkronisering, accesskontroll

<sup>1</sup>lämpligt att demonstrera första passet

# FÖ, SE, LA

- Föreläsningar
  - Slides är *inte* avsedda att fungera självständigt:  
Närvara
  - Ställ frågor
  - Svara (gärna fel) på ställda frågor
- Seminarier
  - Bidra till problemlösning
  - Ställ ännu mer frågor
- Lab
  - Arbeta så långt som möjligt på egen hand
  - Ställ frågor/demonstrera på labpassen
  - Det är ofta hög belastning på passen

## Insiktsnivå

- Jag ställer inga frågor
  - Jag har inte förstått
- Jag ställer grundläggande frågor
  - Jag har förstått att jag inte kan och vill lära mig
- Jag ställer kontrollfrågor
  - Jag har börjat förstå och vill bekräfta det jag kan
- Jag ställer djupa frågor
  - Jag har förstått och vill lära mig mer

## Kommentarer från tidigare år

- I allmänhet uppskattad kurs
- Laborationerna är intressanta, men tar mycket tid
- Mycket information, ibland svårt att veta vad som är "rätt"
- Lite bökigt att testa saker inuti Pintos ibland
- Får ibland vänta länge på labpassen

### Förändringar:

- Mindre förändringar i laborationerna
- Labbar i Teams för bättre flyt
- Mer visualiseringsmöjlighet

- 1 Kursinformation
- 2 Introduktion till C

## C och C++ – vad är skillnaden?

I allmänhet är C ett mycket tunnare lager ovanpå hårdvaran än C++.

Det mesta i C är giltigt i C++, men det finns små skillnader.

Notera att `struct` och `union` har separata namnrymder i C. Går att komma runt med `typedef`.

Notera: `int foo();`  $\not\leftrightarrow$  `int foo(void);`



## C och C++ – vad är skillnaden?

C++	C
cout	printf
klasser	struct
referenser	pekare
vector, string	arrayer
private	definiera i c-fil, static
statisk datamedlem	extern
new, delete	malloc, free
virtuella funktioner	funktionspekare
klassmallar	void *, makron (se Pintos lista)
exceptions	returvärden
överlagring	-

## Utmatning

```
int main() {
    int a = 1;
    unsigned int b = 2;
    float c = 3.5f;
    double d = 5.8;
    char e = 'A';
    int *f = &a;
    const char *g = "hello";

    cout << a << b << c << d
         << e << f << g << endl;
}
```

## Utmatning

```
int main() {
    int a = 1;
    unsigned int b = 2;
    float c = 3.5f;
    double d = 5.8;
    char e = 'A';
    int *f = &a;
    const char *g = "hello";

    printf("%d %u %f %f %c %p %s\n",
           a, b, c, d, e, f, g);
}
```

## Utmatning

```
int main() {
    int a = 10;
    const char *b = "hello";

    printf("%6d\n%06d\n%6s\n", a, a, b);
    // Ger:
    //      10
    // 000010
    //  hello
}
```

## På första labpasset

- C-intro (*Associativ container* används senare)
- Installera Pintos
- (Systemanrop)
  - `exit`, `halt`

Filip Strömbäck, Klas Arvidsson

[www.liu.se](http://www.liu.se)