Velkommen til

INF-1060

Introduksjon til operativsystemer og datakommunikasjon

Forelesere:

- **Pål Halvorsen** (paalh@ifi.uio.no)
  Nettverk og Distribuerte systemer (ND) (ved Simula)
- **Tor Skeie** (tskeie@ifi.uio.no)
  Nettverk og Distribuerte systemer (ND) (ved Simula)
- **Michael Welzl** (michawe@ifi.uio.no)
  Nettverk og Distribuerte systemer (ND) (ved IFI)

Organisasjon:

(alle organisatoriske spørsmål, vedr. oppgaver osv.):
Safiqul Islam (safiquli@ifi.uio.no)
Nettverk og Distribuerte systemer (ND) (ved IFI)
Engelsk foretrekkes!

Pensum-bøker:

- Andrew S. Tanenbaum: Modern Operating Systems, 2008. Pearson. ISBN: 0136006639. 3rd edition.
  http://www.pearsonhighered.com/product?ISBN=0136006639
  ELLER
  William S. Davis and T.M. Rajkumar:
  ”Operating Systems: A Systematic View”, 6. utg.
  Addison-Wesley 2004; ISBN: 0-321-26751-6
  (Tanenbaum foretrekkes)

- Øyvind Hallsteinsen, Bjørn Klefstad og Olav Skundberg: “Innføring i Datakommunikasjon”, Gyldendal Norsk Forlag, 2.utgave 2008. ISBN 9788205384149.

Anbefalt litteratur:

- Brian W. Kernighan, Dennis M. Ritchie: ”The C programming Language”, 2.utgave; Prentice Hall; ISBN 0-13-110362-8
  eller:
- Samuel P. Harbison and Guy L. Steele: ”C: A Reference Manual”, 5. utgave; Prentice Hall 2002; ISBN: 0-13-089592-X

Hjemmeside:

Kursets hjemmeside:
http://www.ifi.uio.no/inf1060/
Er hovedkanalen for løpende informasjon om kurset.

- Lysark til forelesningene blir lagt på hjemmesida
- Ukeoppgaver legges ut hver uke; senere kommer også løsningsforslag. Ukeoppgavene er også pensum!
- Obligatorisk oppgave legges ut på hjemmesida
- ”Hjemme-eksamen” legges ut på hjemmesida
- Viktige beskjeder dukker opp ved behov

Obligatoriske aktiviteter:

Svært lite av kurset er obligatorisk – kun dette:

- Det er obligatorisk frammøte på første forelesning. Frammøte registreres av studadm.
- Det er en obligatorisk oppgave og en hjemme-eksamen som skal løses til fastsatte frister. Den skal leveres individuelt, og det gis karakter på hjemme-eksamenen. Karakteren på den teller 40%, og den avsluttende 4 timers skriftlig eksamen teller 60%.

Følg med på hjemmesida!

Forventninger

Hva kan dere forvente å få ut av å ta kurset?

- Noe kjennskap til programmeringsspråket C og trening i å bruke dette.
- Forstå hva et operativsystem er og hvorledes det fungerer.
- Forstå basale egenskaper ved kommunikasjons-systemer samt kunne skrive enkle programmer som kommuniserer over et nettverk.

Programmeringsspråket C

(Foilene om C er basert på foiler laget av Dag Langmyhr)

Bakgrunn:

- Implementasjon av Unix ved AT&Ts laboratorium i Palo Alto 1960-75.
- Navnet kommer fra BCPL → B → C.
- Opphavsmannen heter *Dennis Ritchie.*
- ANSI-standard i 1989 (”C89”).
- ISO-standard i 1990 – revidert versjon i 1999 (”C99”)
  (ANSI-standard i 2000, men vanligvis ”ANSI C” betyr C89)

Formålet med C:

- Kunne programmere oversiktlig; lettlest kode.
- Tilgang til maskinens ressurser.
- Lite maskinavhengige programmer.
- Kompakte programmer.
- Raske programmer.

Cs fortrinn:

- Mulig å skrive raske programmer.
- Gode muligheter for strukturering av data og program.
- Svært kompakt kode:

Simula	**C**
n:=n+1; A[n]:=A[n]*3.1;	A[++n]*=3.1;

- Mulig å skrive elegante, oversiktlige og portable programmer.
- Fast standard (ANSI C/ISO C) fra 1989.
- Finnes overalt.

Cs svake sider

- Ofte lite portable hvis man ikke tenker på det mens man koder; bedre etter ANSI C.
- C tilbyr programmereren større frihet. Kompilatoren vil dog oppdage færre feil.

Java	C
c = (char)((int)c + 1);	c = c+1;

- Mulighet for kryptisk kode:
  A[*(*x)++=y]+=4;

Sitater:

Å programmere i Java er som å kjøre en Volvo stasjonsvogn; den duver rolig av gårde på veien, men man kommer trygt frem.
Å programmere i C er som å kjøre en Ferrari; den kan gå uhyggelig fort i svingene, men man havner av og til i grøften.

— ukjent opphavsmann

I C er det viktigere at det går fort enn at svaret blir riktig!

— også ukjent opprinnelse

En skrivefeil i C er ingen feil; det er bare et annet program.

— enda en ukjent meningsytrer

Hvorfor er det nyttig å lære C?

Det er flere grunner:

- C er sannsynligvis det mest utbredte språket idag.
- C brukes i et flertall av større programmerings-prosjekter.
- C og Unix er uløselig knyttet sammen.
- Med C kan man skrive raskere kode enn de fleste andre språk.
- Med C kan man skrive svært kompakt kode.
- Programmering i C gir en følelse av hvorledes datamaskinen fungerer.

Et minimalt eksempel:

”Alle” lærebøker i programmering har med følgende lille eksempel:

```
#include <stdio.h>

int main(void)
{
  printf("Hallo, alle sammen!\n");
}
```

(Det var Kernighan & Ritchies første bok om C som startet denne moten!)

I Java ser programmet slik ut:

```
class Hello {
  public static void main(String args[]) {
    System.out.println("Hallo, alle sammen!");
  }
}
```

Kompilering:

Følgende kommandoer kan brukes for å kompilere programmet:

```
cc hallo.c -o hallo
gcc hallo.c -o hallo
```

Det kompilerte programmet kjøres med:

```
hallo
./hallo
```

Alternativ: **clang** - http://clang.llvm.org/

Noen definisjoner:

Program:

Et program er en liste av deklarasjoner av variable og funksjoner:

Java	C
⟨*Klasse-deklarasjoner*⟩	⟨*Deklarasjoner*⟩

Hovedprogrammet:

"Hovedprogrammet" er en funksjon ved navn main:

Java	C
public static void main(. . .) {	int main(void)
⋮	{
}	⋮
	}

Navn:

Store og små bokstaver:

Det er forskjell på store og små bokstaver I C.
MAIN, Main og main er tre helt ulike navn.

Funksjoner:

En C-funksjon ligner veldig på en metode i Java.
Den består alltid av fire deler:

- *type* på returverdien. Hvis ingen returverdi, skrives void.
- *navn* på funksjonen.
- *parameterliste* med typeangivelse av hver parameter. Til forskjell fra Java: hvis det ikke er noen parametre, skrives void.
- *kroppen* som er selve funksjonen. Den er omsluttet av { og }.

  Returverdien angis med en return-setning.

Tekstkonstanter:

Tekstkonstanter skrives med " foran og bak.

Java	C
"En tekst"	"En tekst"

I C kan vi legge inn spesialtegn i teksten; det vanligste er \n som angir linjeskift.

Java	C
"Hei!\n"	"Hei!\n"

Utskrift:

Utskrift skjer via kall på funksjonen printf. Eventuelt linjeskift legges inn i teksten.

Java	C
System.out.print("Hei, ");	printf("Hei, ");
System.out.println("dere!");	printf("dere!\n");

Utskrift av tall

Med %d i teksten kan man angi at det skal settes inn et tall. Dette tallet må komme senere i parameterlisten.

Java	C
System.out.println(a + " og " + b);	printf("%d og %d\n", a, b);

Datatyper i C

I C har vi diverse datatyper som følger:

Navn	Alternativt	Ant byte
signed char	char†	1
unsigned char	char†	1
short	signed short	2
unsigned short		2
int	signed int	2-4
unsigned int	unsigned	2-4
long	signed long	4
unsigned long		4

Vanligvis, men ingen garanti. Alt vi vet er størrelsesrekkefølgen: char <= short <= … etc.

† Standarden sier at det er udefinert om char betyr signed char eller unsigned char så det varierer.

Siden C99 har vi også long long med (vanligvis) 8 byte.

Operatorer:

Aritmetiske operatorer

C har de vanlige aritmetiske operatorene:

+	Addisjon
-	Subtraksjon
*	Multiplikasjon
/	Divisjon
%	Modulo (rest ved divisjon)

Disse kan også brukes til oppdatering av variable:

Koden ...	**... gir det samme som ...**
a += x;	a = a + x;
a -= x;	a = a - x;
⋮	⋮

Sammenligninger:

Sammenligningsoperatorene er også de samme som i Java.

==	Likhet
!=	Ulikhet
<	Mindre enn
<=	Mindre enn eller lik
>	Større enn
>=	Større enn eller lik

Disse operatorene gir 1 om sammenligningen holder og 0 ellers.

NB! Ikke bland sammen = og ==

Logiske verdier/operatorer

Logiske verdier:

Det finnes ingen type boolean i C! I stedet brukes heltall der 0 er **false** og alle andre verdier er **true.**

Logiske operatorer:

! a	1 om a = 0; 0 ellers
a && b	1 om a ≠ 0 og b ≠ 0; 0 ellers
a || b	1 om a ≠ 0 eller b ≠ 0; 0 ellers

Logiske verdier/operatorer (forts.)

Maskeoperatorer:

~	not
&	and
|	or
^	xor

NB! Det er forskjell på logiske og maskeoperatorer!
For eksempel gjelder følgende:

1 && 4 gir 1

1 & 4 gir 0