

TDIU11 – Föreläsning 1

Introduktion till operativsystem

Filip Strömbäck

- 1 **Introduktion**
- 2 Kursinformation
- 3 Operativsystem
- 4 Multiprogrammering

Överblick

Bild från htop:

```

0.6% Tasks: 57, 33 thr, 75 kthr; 1 running
1.0% Load average: 0.00 0.00 0.00
Mem[|||||]294M/3.82G Uptime: 90 days, 12:22:18
Swp[|||||]482M/1.02G

Main 100%

PIDUSER      PRI NI  VIRT  RES  SHR  S  CPU% MEM%  TIME+  Command
1 root        20  0  164M  8020  6396  S  0.0  0.2  14:26.84 /lib/systemd/systemd --system --c
420 root        20  0  5868  1732  1572  S  0.0  0.0  0:00.04 dhclient -4 -v -i -pf /run/dhc
451 root        20  0  9484  1628  1496  S  0.0  0.0  0:22.73 /usr/sbin/cron -f
452 messagebus  20  0  9428  3260  2480  S  0.0  0.1  1:49.11 /usr/bin/dbus-daemon --system
461 root        20  0  17288  3764  2908  S  0.0  0.1  0:56.57 /lib/systemd/systemd-logind
464 root        20  0  8748  608  524  S  0.0  0.0  0:00.00 /sbin/agetty -o -p -- \u --noc
539 mysql      20  0  1447M 18268  6580  S  0.0  0.5  24:29.02 /usr/sbin/mariadb
779 mysql      20  0  1447M 18268  6580  S  0.0  0.5  17:54.95 /usr/sbin/mariadb
800 mysql      20  0  1447M 18268  6580  S  0.0  0.5  0:09.35 /usr/sbin/mariadb
805 mysql      20  0  1447M 18268  6580  S  0.0  0.5  0:00.00 /usr/sbin/mariadb
808 mysql      20  0  1447M 18268  6580  S  0.0  0.5  0:00.00 /usr/sbin/mariadb
834 mysql      20  0  1447M 18268  6580  S  0.0  0.5  0:00.00 /usr/sbin/mariadb
836 mysql      20  0  1447M 18268  6580  S  0.0  0.5  0:00.00 /usr/sbin/mariadb
1595631 mysql  20  0  1447M 18268  6580  S  0.0  0.5  0:00.12 /usr/sbin/mariadb
1595897 mysql  20  0  1447M 18268  6580  S  0.0  0.5  0:00.00 /usr/sbin/mariadb
1595901 mysql  20  0  1447M 18268  6580  S  0.0  0.5  0:00.00 /usr/sbin/mariadb
644 root        20  0  5072  2196  1704  S  0.0  0.1  0:01.97 dhclient -cf /etc/dhcp/dhclient
766 root        20  -20 3428  884  704  S  0.0  0.0  0:00.18 /usr/sbin/ntpd -f /etc/ntp.conf
767 ntpd         20  -20 3580  2004  1688  S  0.0  0.1  0:52.72 ntpd: ntp engine
768 ntpd         20  0  3508  1756  1544  S  0.0  0.0  0:00.00 ntpd: dns engine
32868 filip     20  0  23476  5616  4060  S  0.0  0.1  0:00.29 /usr/bin/python3 /home/filip/s
33867 filip     20  0  23604  9900  3280  S  0.0  0.2  0:00.32 /usr/bin/python3 /home/filip/s
151697 postfixed 20  0  2488  1468  1352  S  0.0  0.0  4:49.62 /usr/sbin/postfixd -f 30001 -t
154699 bind       20  0  446K  9572  5160  S  0.0  0.2  7:36.28 /usr/sbin/named -f -u bind
154700 bind       20  0  446K  9572  5160  S  0.0  0.2  1:06.06 /usr/sbin/named -f -u bind
154701 bind       20  0  446K  9572  5160  S  0.0  0.2  0:01.42 /usr/sbin/named -f -u bind
154702 bind       20  0  446K  9572  5160  S  0.0  0.2  6:28.62 /usr/sbin/named -f -u bind
154708 bind       20  0  446K  9572  5160  S  0.0  0.2  0:00.04 /usr/sbin/named -f -u bind
154709 bind       20  0  446K  9572  5160  S  0.0  0.2  0:00.04 /usr/sbin/named -f -u bind

```

Överblick

Bild från htop:

0.6% Tasks: 57, 33 thr, 75 kthr; 1 running
 2.0% Load average: 0.00 0.00 0.00
 Mem[|||||] 294M/3.82G Uptime: 90 days, 12:22:18
 Swp[|||||] 482M/1.02G

PID	USER	PR	NI	VIRT	RES	SHR	S	CPUS	MEM	TIME	Command
1	root	0	0	164M	802K	639K	S	0.0	0.2	14:26.84	/lib/systemd/systemd --system --
420	root	20	0	586K	1732	1572	S	0.0	0.0	0:00.04	dhclient -4 -v -i -p /run/dhc
451	root	20	0	948K	1628	1496	S	0.0	0.0	0:22.73	/usr/sbin/cron -f
452	messagebus	20	0	942K	3260	2480	S	0.0	0.1	1:49.11	/usr/bin/dbus-daemon --system
461	root	20	0	1728K	3764	2908	S	0.0	0.1	0:56.57	/lib/systemd/systemd-logind
464	root	20	0	874K	608	524	S	0.0	0.0	0:00.00	/sbin/agetty -o -p -- \u --noc
539	mysqld	20	0	1447K	1826K	6580	S	0.0	0.5	24:29.02	/usr/sbin/mariadb
779	mysqld	20	0	1447K	1826K	6580	S	0.0	0.5	17:54.95	/usr/sbin/mariadb
800	mysqld	20	0	1447K	1826K	6580	S	0.0	0.5	0:09.35	/usr/sbin/mariadb
805	mysqld	20	0	1447K	1826K	6580	S	0.0	0.5	0:00.00	/usr/sbin/mariadb
808	mysqld	20	0	1447K	1826K	6580	S	0.0	0.5	0:00.00	/usr/sbin/mariadb
834	mysqld	20	0	1447K	1826K	6580	S	0.0	0.5	0:00.00	/usr/sbin/mariadb
836	mysqld	20	0	1447K	1826K	6580	S	0.0	0.5	0:00.00	/usr/sbin/mariadb
1595631	mysqld	20	0	1447K	1826K	6580	S	0.0	0.5	0:00.12	/usr/sbin/mariadb
1595897	mysqld	20	0	1447K	1826K	6580	S	0.0	0.5	0:00.00	/usr/sbin/mariadb
1595901	mysqld	20	0	1447K	1826K	6580	S	0.0	0.5	0:00.00	/usr/sbin/mariadb
644	root	20	0	5072	2196	1704	S	0.0	0.1	0:01.97	dhclient -cf /etc/dhcp/dhclient
766	root	0	-20	342K	884	704	S	0.0	0.0	0:00.18	/usr/sbin/ntpd -f /etc/openntp
767	ntpd	0	-20	3580	2004	1688	S	0.0	0.1	0:52.72	ntpd: ntp engine
768	ntpd	20	0	350K	1756	1544	S	0.0	0.0	0:00.00	ntpd: dns engine
32868	filip	20	0	2347K	5616	4060	S	0.0	0.1	0:00.29	/usr/bin/python3 /home/filip/s
33867	filip	20	0	2360K	9900	3280	S	0.0	0.2	0:00.32	/usr/bin/python3 /home/filip/s
151697	postfix	20	0	248K	146K	132K	S	0.0	0.0	4:49.62	/usr/sbin/postfixd -f 30001 -t
154699	bind	20	0	446K	9572	5160	S	0.0	0.2	7:36.28	/usr/sbin/named -f -u bind
154700	bind	20	0	446K	9572	5160	S	0.0	0.2	1:06.06	/usr/sbin/named -f -u bind
154701	bind	20	0	446K	9572	5160	S	0.0	0.2	0:01.42	/usr/sbin/named -f -u bind
154702	bind	20	0	446K	9572	5160	S	0.0	0.2	6:28.62	/usr/sbin/named -f -u bind
154708	bind	20	0	446K	9572	5160	S	0.0	0.2	0:00.04	/usr/sbin/named -f -u bind
154709	bind	20	0	446K	9572	5160	S	0.0	0.2	0:00.04	/usr/sbin/named -f -u bind

Intressanta frågor:

- Vi har 2 CPU-kärnor, men 57 "tasks". Hur fungerar det?
- Hur ser vi till att olika program kan samarbeta utan att förstöra för varandra?

Överblick

Bild från htop:

System statistics from htop:

- Tasks: 57, 33 thr, 75 kthr; 1 running
- Load average: 0.00 0.00 0.00
- Mem: 294M/3.82G
- Uptime: 90 days, 12:22:18
- Swp: 482M/1.02G

Process list (PID, USER, PRI, NI, VIRT, RES, SHR, S, CPU%, MEM%, TIME+, Command):

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
1	root	20	0	164M	8020	6396	S	0.0	0.2	14:26.84	/lib/systemd/systemd --system --c
420	root	20	0	5868	1732	1572	S	0.0	0.0	0:00.04	dhclient -4 -v -i -pf /run/dhc
451	root	20	0	9484	1628	1496	S	0.0	0.0	0:22.73	/usr/sbin/cron -f
452	messagebus	20	0	9428	3260	2480	S	0.0	0.1	1:49.11	/usr/bin/dbus-daemon --system
461	root	20	0	17288	3764	2908	S	0.0	0.1	0:56.57	/lib/systemd/systemd-logind
464	root	20	0	8748	608	524	S	0.0	0.0	0:00.00	/sbin/agetty -o -p -- /u --noc
539	mysqld	20	0	1447M	18268	6580	S	0.0	0.5	24:29.02	/usr/sbin/mariadb
779	mysqld	20	0	1447M	18268	6580	S	0.0	0.5	17:54.95	/usr/sbin/mariadb
800	mysqld	20	0	1447M	18268	6580	S	0.0	0.5	0:09.35	/usr/sbin/mariadb
805	mysqld	20	0	1447M	18268	6580	S	0.0	0.5	0:00.00	/usr/sbin/mariadb
808	mysqld	20	0	1447M	18268	6580	S	0.0	0.5	0:00.00	/usr/sbin/mariadb
834	mysqld	20	0	1447M	18268	6580	S	0.0	0.5	0:00.00	/usr/sbin/mariadb
836	mysqld	20	0	1447M	18268	6580	S	0.0	0.5	0:00.00	/usr/sbin/mariadb
1595631	mysqld	20	0	1447M	18268	6580	S	0.0	0.5	0:00.12	/usr/sbin/mariadb
1595897	mysqld	20	0	1447M	18268	6580	S	0.0	0.5	0:00.00	/usr/sbin/mariadb
1595901	mysqld	20	0	1447M	18268	6580	S	0.0	0.5	0:00.00	/usr/sbin/mariadb
644	root	20	0	5072	2196	1704	S	0.0	0.1	0:01.97	dhclient -cf /etc/dhcp/dhclient
766	root	20	-20	3428	884	704	S	0.0	0.0	0:00.18	/usr/sbin/ntpd -f /etc/openntp
767	ntpd	20	-20	3580	2004	1688	S	0.0	0.1	0:52.72	ntpd: ntp engine
768	ntpd	20	0	3508	1756	1544	S	0.0	0.0	0:00.00	ntpd: dns engine
32868	filip	20	0	23476	5616	4060	S	0.0	0.1	0:00.29	/usr/bin/python3 /home/filip/s
33867	filip	20	0	23604	9900	3280	S	0.0	0.2	0:00.32	/usr/bin/python3 /home/filip/s
151697	postfixrad	20	0	2488	1468	1352	S	0.0	0.0	4:49.62	/usr/sbin/postarad -f 20001 -t
154699	bind	20	0	446K	9572	5160	S	0.0	0.2	7:36.28	/usr/sbin/named -f -u bind
154700	bind	20	0	446K	9572	5160	S	0.0	0.2	1:06.06	/usr/sbin/named -f -u bind
154701	bind	20	0	446K	9572	5160	S	0.0	0.2	0:01.42	/usr/sbin/named -f -u bind
154702	bind	20	0	446K	9572	5160	S	0.0	0.2	6:28.62	/usr/sbin/named -f -u bind
154708	bind	20	0	446K	9572	5160	S	0.0	0.2	0:00.04	/usr/sbin/named -f -u bind
154709	bind	20	0	446K	9572	5160	S	0.0	0.2	0:00.04	/usr/sbin/named -f -u bind

Intressanta frågor:

- Varför ser "mem" full ut, trots att bara 294 MiB av 3,82 GiB används?
- Varför använder systemd 164 MiB "virt", men bara 9 MiB "res" och 6 MiB "shr"?
- Vad innebär "swp"? När används den?

Överblick

Bild från htop:

System statistics from htop:

- Tasks: 57, 33 thr, 75 kthr; 1 running
- Load average: 0.00 0.00 0.00
- Mem: 294M/3.82G
- Uptime: 90 days, 12:22:18
- Swap: 482M/1.02G

PID	USER	PR	NI	VIRT	RES	SHR	S	CPUS	MEM	TIME	Command
1	root	20	0	164M	8020	6396	S	0.0	0.2	14:26.84	/lib/systemd/systemd --system --
420	root	20	0	5868	1732	1572	S	0.0	0.0	0:00.04	dhclient -4 -v -i -p /run/dhc
451	root	20	0	9484	1628	1496	S	0.0	0.0	0:22.73	/usr/sbin/cron -f
452	messagebus	20	0	9428	3260	2480	S	0.0	0.1	1:49.11	/usr/bin/dbus-daemon --system
461	root	20	0	17288	3764	2908	S	0.0	0.1	0:56.57	/lib/systemd/systemd-logind
464	root	20	0	8748	608	524	S	0.0	0.0	0:00.00	/sbin/agetty -o -p -- \u --noc
539	mysql	20	0	1447M	18268	6580	S	0.0	0.5	24:29.02	/usr/sbin/mariadb
779	mysql	20	0	1447M	18268	6580	S	0.0	0.5	17:54.95	/usr/sbin/mariadb
800	mysql	20	0	1447M	18268	6580	S	0.0	0.5	0:09.35	/usr/sbin/mariadb
805	mysql	20	0	1447M	18268	6580	S	0.0	0.5	0:00.00	/usr/sbin/mariadb
808	mysql	20	0	1447M	18268	6580	S	0.0	0.5	0:00.00	/usr/sbin/mariadb
834	mysql	20	0	1447M	18268	6580	S	0.0	0.5	0:00.00	/usr/sbin/mariadb
836	mysql	20	0	1447M	18268	6580	S	0.0	0.5	0:00.00	/usr/sbin/mariadb
1595631	mysql	20	0	1447M	18268	6580	S	0.0	0.5	0:00.12	/usr/sbin/mariadb
1595897	mysql	20	0	1447M	18268	6580	S	0.0	0.5	0:00.00	/usr/sbin/mariadb
1595901	mysql	20	0	1447M	18268	6580	S	0.0	0.5	0:00.00	/usr/sbin/mariadb
644	root	20	0	5072	2196	1704	S	0.0	0.1	0:01.97	dhclient -cf /etc/dhcp/dhclient
766	root	20	-20	3428	884	704	S	0.0	0.0	0:00.18	/usr/sbin/ntpd -f /etc/openntp
767	ntpd	20	-20	3580	2004	1688	S	0.0	0.1	0:52.72	ntpd: ntp engine
768	ntpd	20	0	3508	1756	1544	S	0.0	0.0	0:00.00	ntpd: dns engine
32868	filip	20	0	23476	5616	4060	S	0.0	0.1	0:00.29	/usr/bin/python3 /home/filip/s
33867	filip	20	0	23604	9900	3280	S	0.0	0.2	0:00.32	/usr/bin/python3 /home/filip/s
151697	postfix	20	0	2488	1468	1352	S	0.0	0.0	4:49.62	/usr/sbin/postfix -f 30001 -t
154699	bind	20	0	446K	9572	5160	S	0.0	0.2	7:36.28	/usr/sbin/named -f -u bind
154700	bind	20	0	446K	9572	5160	S	0.0	0.2	1:06.06	/usr/sbin/named -f -u bind
154701	bind	20	0	446K	9572	5160	S	0.0	0.2	0:01.42	/usr/sbin/named -f -u bind
154702	bind	20	0	446K	9572	5160	S	0.0	0.2	6:28.62	/usr/sbin/named -f -u bind
154708	bind	20	0	446K	9572	5160	S	0.0	0.2	0:00.04	/usr/sbin/named -f -u bind
154709	bind	20	0	446K	9572	5160	S	0.0	0.2	0:00.04	/usr/sbin/named -f -u bind

Intressanta frågor:

- Hur lagrar vi olika filer på samma disk?
- Hur ser vi till att bara rätt program kan komma åt rätt information?

Mål med kursen

Få insyn i...

- ...*vad* ett operativsystem är
- ...*vad* det gör
- ...*varför* det behövs

Mål med kursen

Få insyn i...

- ...*vad* ett operativsystem är
- ...*vad* det gör
- ...*varför* det behövs

Så att vi kan...

- ...*förstå* hur våra program körs
- ...*förstå* de abstraktioner som OS introducerar
- ...*utnyttja* abstraktionerna effektivt
- ...*undvika* begränsningar som finns

- 1 Einführung
- 2 **Kursinformation**
- 3 Betriebssystem
- 4 Multiprogrammierung

Resurser

- Kurshemsida: <https://www.ida.liu.se/~TDIU11/>
- Litteratur: Operating System Concepts
- Kod för Pintos: <https://gitlab.liu.se/tdiu16-material/pintos>

Kursledare	Filip Strömbäck
Kursassistent	Christoffer Holm
Assistenter	Eric Ekström
	Malte Nilsson
	Malte Hammar
Administratör	Emilie Lind

Examination

UPG1 Problemlösning, 3 hp (U, G)

Förbereda och diskutera inlämningsuppgifter (*utmaningar*) till de fem tillfällena som finns under kursens gång.

UPG2 Inlämningsuppgift, 1 hp (U, G)

Läsa artiklar och skriva en sammanfattande rapport.

TEN1 Skriftlig tentamen, 2 hp (U, 3, 4, 5)

Skriftlig papperstentamen, liknande UPG1.

Tidigare år

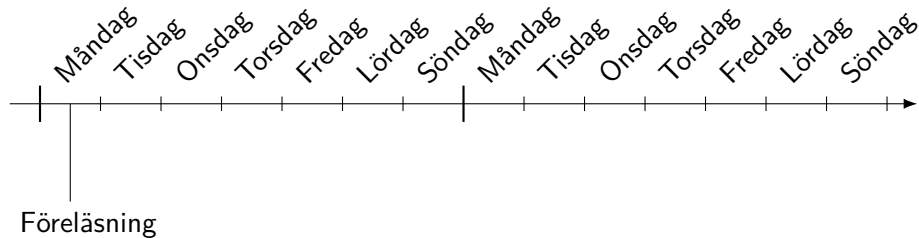
Kommentarer

- Seminarier: väldigt givande, men tar tid

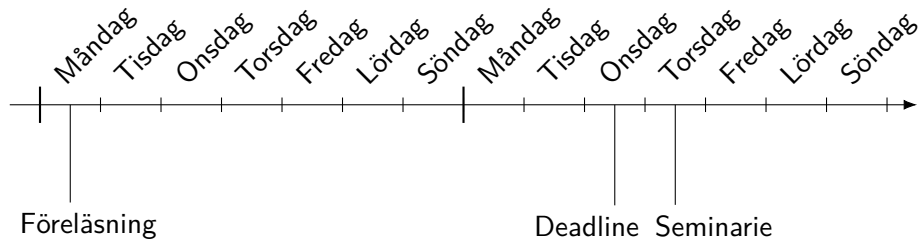
Nytt för i år

- Nya rutiner kring inlämning (i Lisam)

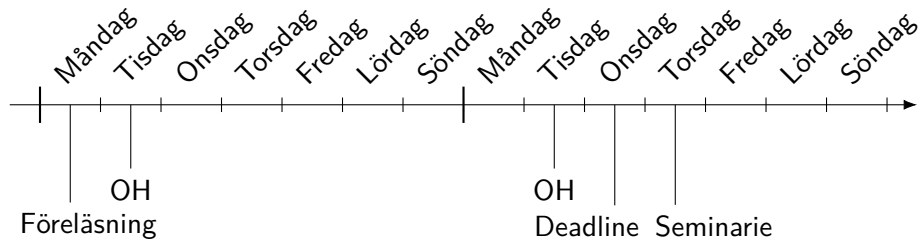
Struktur



Struktur

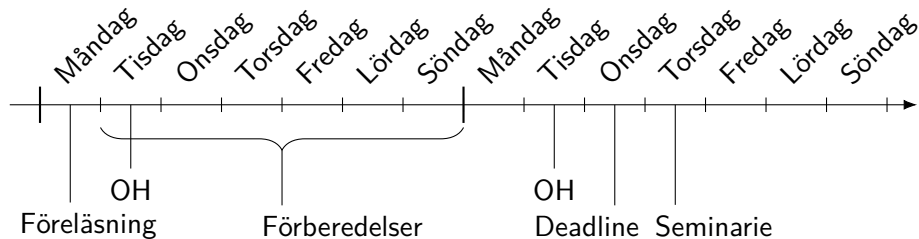


Struktur



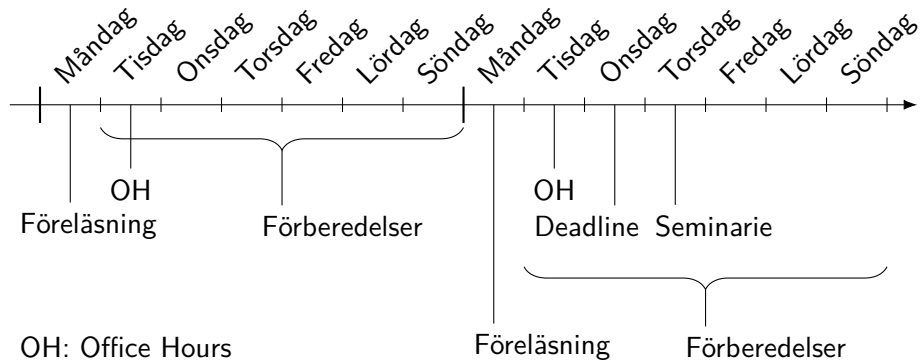
OH: Office Hours

Struktur



OH: Office Hours

Struktur



Föreläsningar

- Måndagar (i de flesta fall)
- Introducerar teori till kommande seminarier
- Lëshänvisningar till kursboken finns på kurshemsidan
- Skumma gärna igenom boken innan föreläsningen – ställ frågor på sådant som är oklart
- Slides finns på kurshemsidan – som anteckningsstöd, inte gjorda för att gå att läsa utan att ha närvarat på föreläsningen

Office Hours

- Tisdagar (förutom första veckan och veckan med LARM)
- Möjlighet att ställa frågor relaterade till kursen, diskutera lösningar, etc.
- (Går självklart att skicka e-post också)
- Nytt för i år, vi utvärderar hur de nyttjas

Seminarier: Utmaningar (UPG1)

- 5 tillfällen
- 8 problem/tillfälle
- 40 problem totalt
- 16 lösta problem ger G
- Var 4:e löst problem utöver detta ger 1 p på mot högre betyg på tentan

Förberedelse:

- Lös en delmängd av problemen
- Skicka in lösningsanteckningar via Lisam
Viktigt: Skriv *Jag har löst uppgift: 1, 3, 5, 7* i textrutan vid inlämning
- Samarbete är OK och uppmuntrat. Skicka in dina *egna* lösningsanteckningar, var beredd att presentera.

Seminarier: Utmaningar (UPG1)

Under seminariet:

- 5 tillfällen
- 8 problem/tillfälle
- 40 problem totalt
- 16 lösta problem ger G
- Var 4:e löst problem utöver detta ger 1 p på mot högre betyg på tentan

- Assistenten väljer studenter slumpvis
- Presentation utifrån medtagna anteckningar
- Diskussion av lösningen
- Kom gärna även om du inte löst något problem!

Notera: Lösningen behöver *inte* vara helt korrekt. Visar du att du har förstått det som krävs för att lösa problemet är det OK. Kan du inte det förlorar du *alla* poäng från seminariet.

Seminarier: Rapporter (UPG2)

- 2 ordinare tillfällen
- 1 artikel/tillfälle
- Se kurshemsidan för info om komplettering

Inför seminariet:

- Läs och sammanfatta en OS-relaterad artikel
- Skicka in via Lisam innan seminariet, **samt** skriv ut och ta med till seminariet

Under seminariet:

- Läs och ge feedback till en annan student
- Diskussion i större grupp

Seminarier: Rapporter (UPG2)

Rapport 1:

- 2 ordinare tillfällen
 - 1 artikel/tillfälle
 - Se kurshemsidan för info om komplettering
- Fokusera på helheten i artikeln
 - Diskutera detta på seminariet
 - Arbetet här underlättar rapport 2

Rapport 2:

- Bygger vidare på rapport 1
- Nyare version av innehållet i rapport 1
- Assistent ger feedback på rapporten efteråt

Planering

Vecka	Föreläsning	Seminarie
4	Processer och trådar	—
5	Filsystem och lagring	Utmaning 1: Schemaläggning
6	Minneshantering	Rapport 1: Filsystem I
7	Virtuellt minne	Utmaning 2: Filsystem
8	Säkerhet	Utmaning 3: Virtuellt minne
9	Repetition/utblickar	Rapport 2: Filsystem II
10	—	Utmaning 4: Säkerhet
11	Tentaförberedelse	Utmaning 5: Repetition (tisdag)
12	(omtenta-p)	Rapport: Uppsamling

Planering

- Finns material att arbeta med varje vecka
- Finns däremot flexibilitet vilka seminarier du fokuserar på
- Planera i förväg för sammanfattningarna – bra idé att läsa igenom artikeln en första gång i god tid!

Webreg

Anmäl er till en grupp i Webreg

- Finns två moment, en för UPG1 och en för UPG2.
- Anmäl er i båda

Notera: Deadline är den 21:a — grund för grupper i Lisam för inlämning.

- 1 Einführung
- 2 Kursinformation
- 3 **Operativsystem**
- 4 Multiprogrammierung

Varför vill vi ha ett operativsystem?

Varför vill vi ha ett operativsystem?

Användbarhet:

- Användare förväntar sig viss funktionalitet
- Programmerare vill inte behöva implementera filsystem/IO/etc.

Portabilitet:

- Vi vill att program ska fungera på olika datorer med olika hårdvara
- Vill inte behöva implementera stöd för all möjlig hårdvara i varje program

Varför vill vi ha ett operativsystem?

Effektivitet:

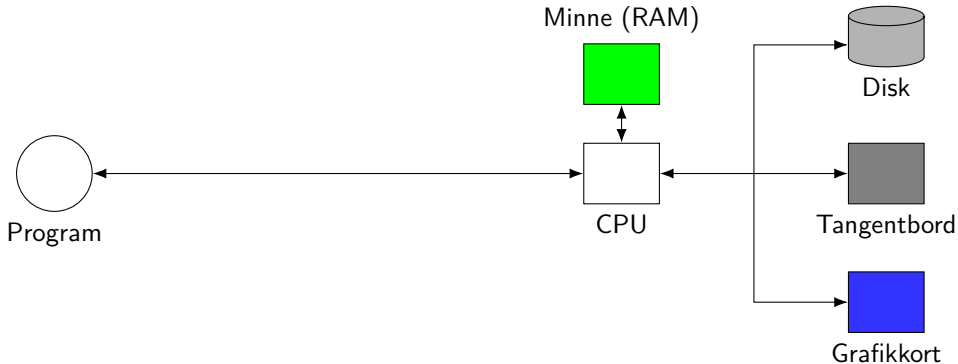
- Vi vill nyttja hårdvaran effektivt:
 - ⇒ Vi vill kunna köra flera program samtidigt
 - ⇒ Vi vill kunna låta flera användare använda systemet samtidigt

Säkerhet:

- Program vi kör samtidigt ska inte kunna förstöra för varandra
- Program ska inte kunna förstöra för OS
- Respektera användares begränsningar i åtkomst av data

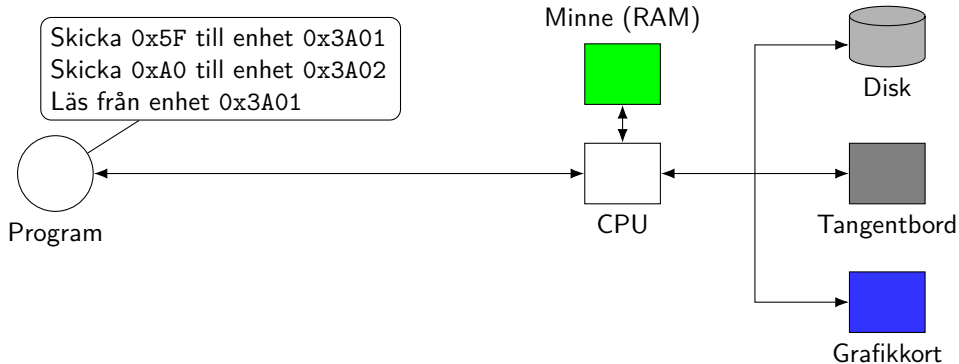
Översikt

Utan OS



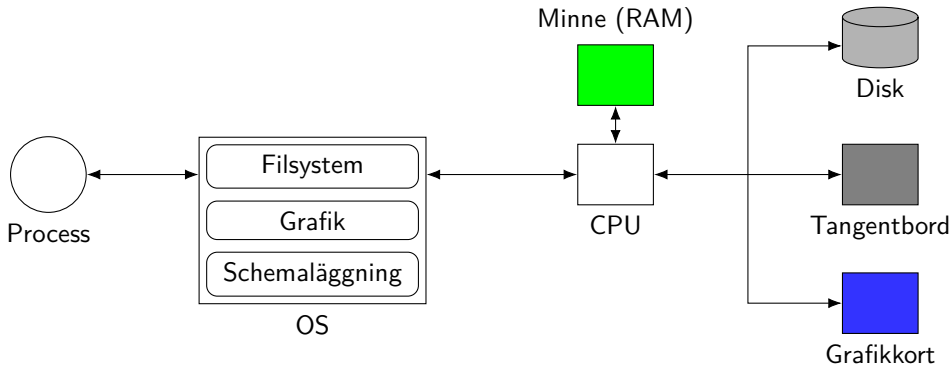
Översikt

Utan OS

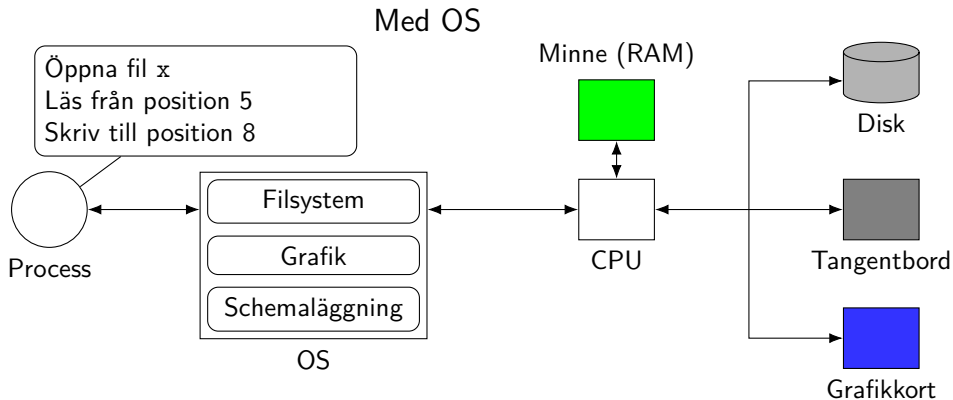


Översikt

Med OS



Översikt



Vad består ett operativsystem av?

- **Kernel/kärna** (fokus i kursen)
Delen av OS som *alltid* körs, koordinerar processer som körs, ger abstraktion för hårdvara, etc.
- **Användargränssnitt** (skal, shell)
Program för att interagera med användaren
Ex. vis: bash, terminal, fönsterhanterare, skrivbordsmiljö
- **Verktyg**
Program som hjälper användaren att göra saker
Ex. vis: cp, g++, emacs, libreoffice, firefox

Var går gränsen för vad som är ett operativsystem?

Olika typer av operativsystem

- Single-user
 - Äldre system, tillåter oftast bara att ett program körs samtidigt
 - OS kan helt enkelt vara en samling funktioner som kan anropas
 - Tillåter ibland att program kringgår OS och pratar direkt med hårdvaran
 - Exempel: DOS
- Batchsystem
- Time-sharing
- Real-time

Olika typer av operativsystem

- Single-user
- Batchsystem
 - Tidiga operativsystem – stordator i källaren
 - Byggda för att hantera *job* effektivt
 - Kör flera job samtidigt för att nyttja systemet effektivt
 - Inte byggt för interaktiv användning, svarstider kan vara långa
 - Nutida exempel: Superdatorn vid NSC
- Time-sharing
- Real-time

Olika typer av operativsystem

- Single-user
- Batchsystem
- Time-sharing
 - Moderna operativsystem
 - Kör flera processer "samtidigt", ibland åt olika användare
 - Har som mål att användas interaktivt, och strävar efter att låta alla processer köras relativt ofta
 - Exempel: Linux, Windows, MacOS, ...
- Real-time

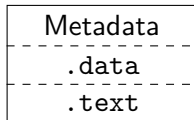
Olika typer av operativsystem

- Single-user
- Batchsystem
- Time-sharing
- Real-time
 - Ofta i inbyggda system
 - Har likt time-sharing stöd för flera processer
 - Ger *garantier* på när systemet svarar på externa händelser
 - Finns *soft* real-time och *hard* real-time
 - Exempel: FreeRTOS, (real-time Linux)

- 1 Einführung
- 2 Kursinformation
- 3 Betriebssystem
- 4 Multiprogrammierung

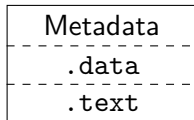
Processer och trådar

Program (fil)

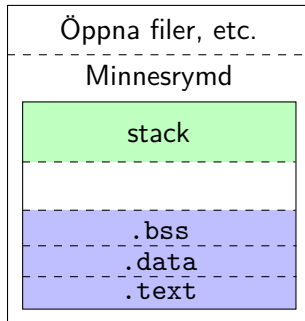


Processer och trådar

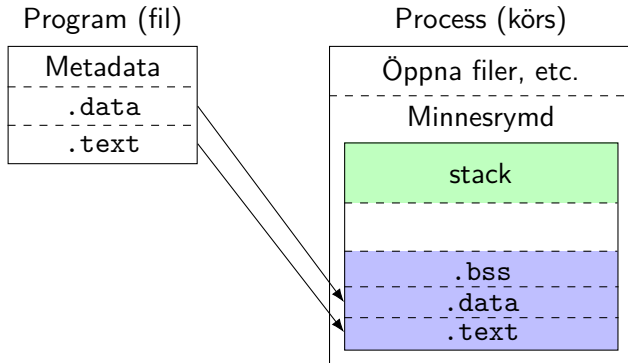
Program (fil)



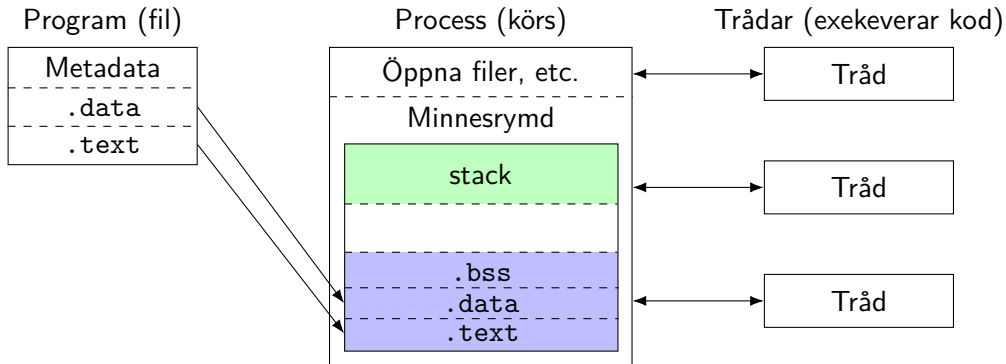
Process (körs)



Processer och trådar



Processer och trådar



Hur hanterar vi flera processer på ett säkert sätt?

Hårdvaran ger oss:

- Dual-mode exekevering
- Virtuellt minne
- Avbrott (Interrupt)
- Timers

Dual-mode

Mål: Vill kunna "låsa ner" CPU när vi kör kod utanför kernel

Lösning: vi har en "mode-bit" i CPU (eller motsvarande)

Exempel: x86 har olika "ringar" som får göra olika mycket med systemet:

0. Kernel mode
1. (används ej)
2. (används ej)
3. User mode

Dual-mode

x86 har olika "ringar" som får göra olika mycket med systemet:

0. Kernel mode
1. (används ej)
2. (används ej)
3. User mode

Endast ring 0 får köra följande instruktioner:

- `hlt` Halt – avbryter exekevering tills interrupt kommer
- `lgdt` Load global descriptor table – ändrar hur virtuellt minne ser ut
- `in, out` I/O till hårdvara

Dual-mode

x86 har olika "ringar" som får göra olika mycket med systemet:

- 2. System management
- 1. Hypervisor
- 0. Kernel mode
- 1. (används ej)
- 2. (används ej)
- 3. User mode

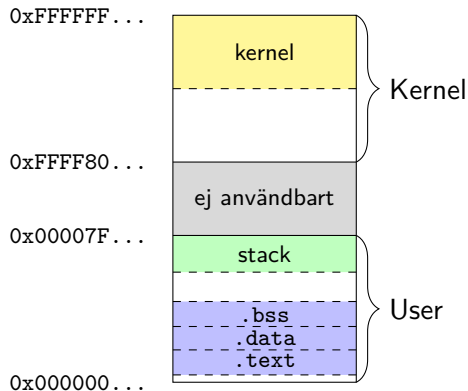
Endast ring 0 eller lägre får köra följande instruktioner:

- `hlt` Halt – avbryter
exekvering tills interrupt
kommer
- `lgdt` Load global descriptor
table – ändrar hur
virtuellt minne ser ut
- `in, out` I/O till hårdvara

Virtuellt minne

Vi vill också förhindra att processer kommer åt varandras och kernels minne

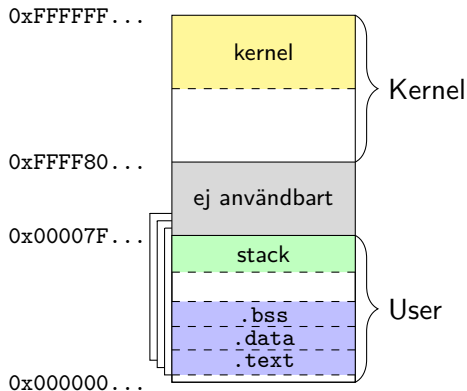
- CPU tillhandahåller *paging*
- Vi kan implementera *virtuellt minne*
- Varje process får egen vy av RAM
- Kan bara komma åt minne de "borde" kunna komma åt



Virtuellt minne

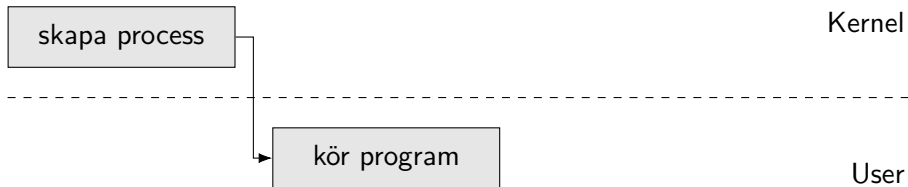
Vi vill också förhindra att processer kommer åt varandras och kernels minne

- CPU tillhandahåller *paging*
- Vi kan implementera *virtuellt minne*
- Varje process får egen vy av RAM
- Kan bara komma åt minne de "borde" kunna komma åt



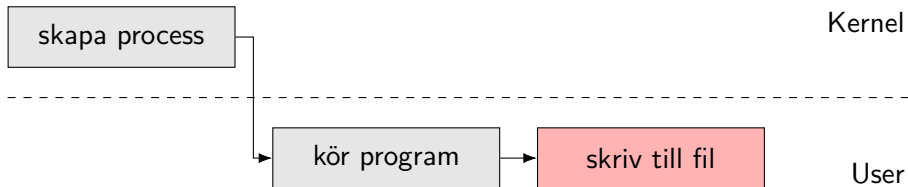
Problem så här långt

Vad vi kan göra nu:



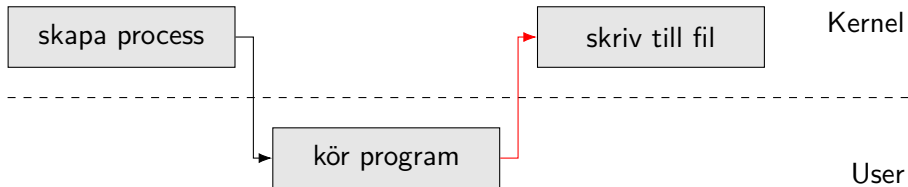
Problem så här långt

Vad vi kan göra nu:



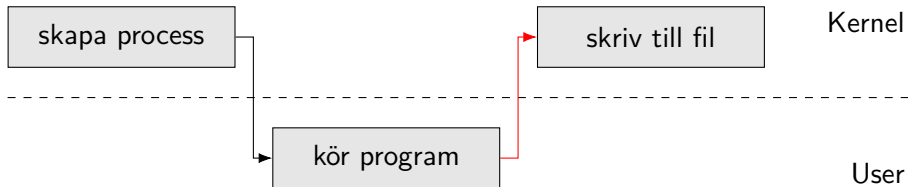
Problem så här långt

Vad vi kan göra nu:



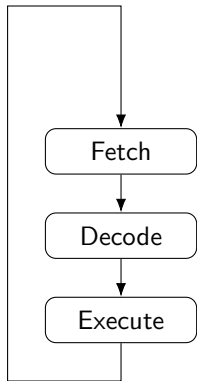
Problem så här långt

Vad vi kan göra nu:

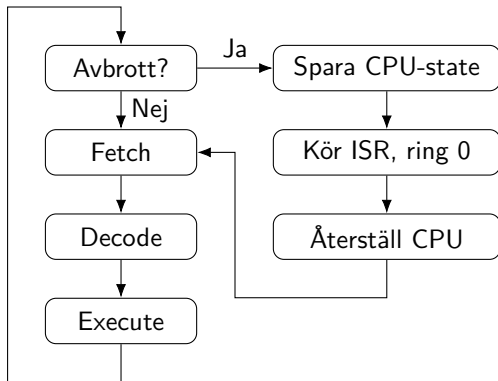


Hur löser vi detta?

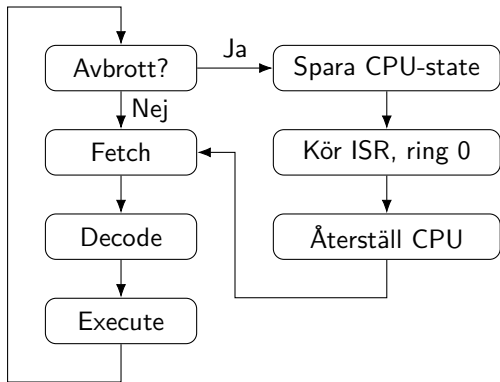
Avbrott (interrupt)



Avbrott (interrupt)



Avbrott (interrupt)



- ISR specificeras av *kernel*
- ISR körs som *kernel*
- Genereras av *hårdvara* eller *mjukvara*
- Kan användas för att *säkert* återgå till högre privilegier

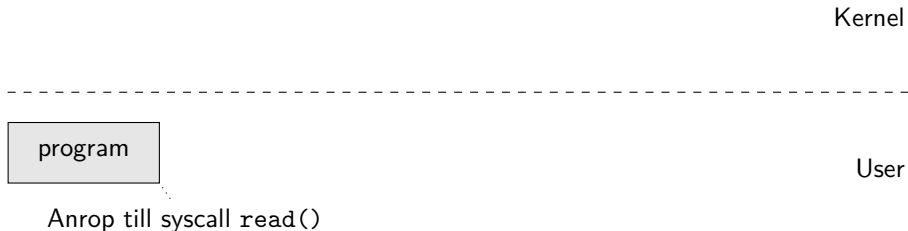
Systemanrop

Vi kan använda interrupt för att implementera *systemanrop*:



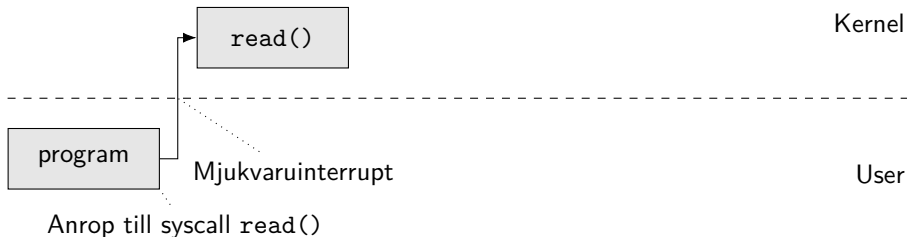
Systemanrop

Vi kan använda interrupt för att implementera *systemanrop*:



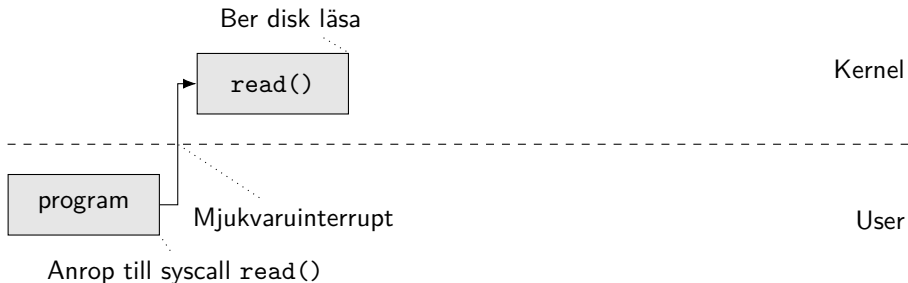
Systemanrop

Vi kan använda interrupt för att implementera *systemanrop*:



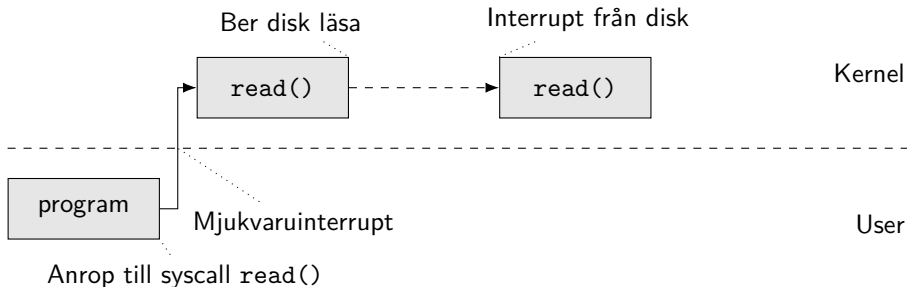
Systemanrop

Vi kan använda interrupt för att implementera *systemanrop*:



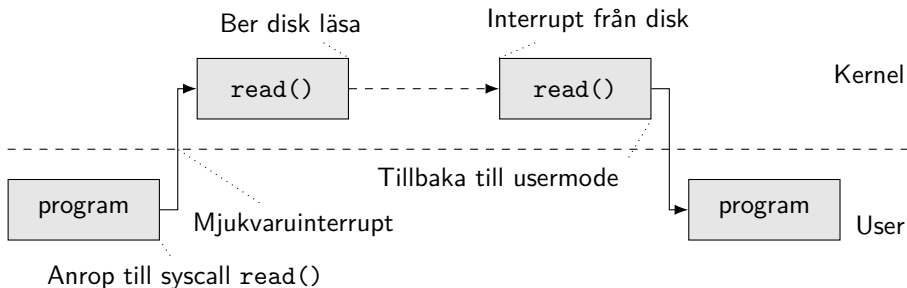
Systemanrop

Vi kan använda interrupt för att implementera *systemanrop*:



Systemanrop

Vi kan använda interrupt för att implementera *systemanrop*:



Filip Strömbäck

www.liu.se