

Principper for Samtidighed og Styresystemer

Interrupts og Styresystemkernen

René Rydhof Hansen

April 2008

- Opgave 1: Sideerstatningsalgoritmer
- Opgave 2: Gamle opgaver (hukommelseshåndtering)
- Opgave 3: Logisk adresse til fysisk adresse (Pentium)
- Opgave 4: Global vs. lokal sideerstatningsalgoritme
- Opgave 5: Mutex instruktioner
- Opgave 6: Priviligerede instruktioner
- Opgave 7: Exceptions

Pop-Up Test #1: Resultat

- Designkriterier: nem og hurtig...
- Korrekt besvarelse
- Statistik

Pop-Up Test #1: Resultat

- Designkriterier: nem og hurtig... **men ikke så nem og hurtig!**
- Korrekt besvarelse
- Statistik

- At kunne redegøre for brugen af “user-” og “kernelmode”.
- At kunne definere og redegøre for generering og håndtering af “exceptions” (herunder interrupts).
- At kunne redegøre for forskellige former for I/O (herunder interruptstyret).
- At kunne redegøre for håndtering af systemkald.
- At kunne redegøre for tråd- og procesrealisering (herunder fem-tilstandsmodellen)
- At kunne redegøre for scheduling: begreb og brug

User- og Kernelmode

- Behov for at sikre dele af styresystemet mod “andre”
- Sikring af privilegerede instruktioner
 - Instruktioner der kan påvirke andre processer negativt
 - Standsning af processoren, mode-change, deaktivering af interrupts, ...
- User-mode: til “almindelige” programmer
- Kernel-mode: til styresystemet

Example (Pentium)

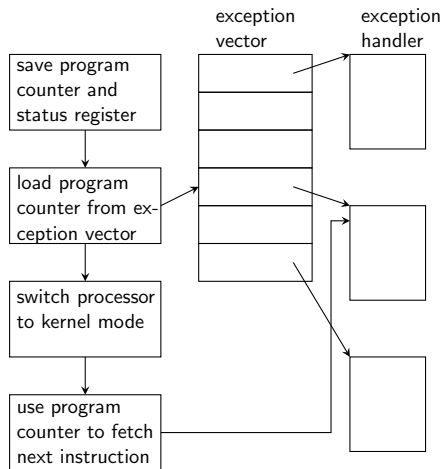
Fire modes (kaldet *rings*): 0–3. Windows og Linux bruger kun de to.

Exceptions

- Exceptions genereres af CPU'en som følge af en hændelse i eller udenfor CPU'en
- Eksempler
 - Faults
 - Division by zero
 - Brug af illegale instruktioner
 - Utilstrækkelige rettigheder
 - Traps
 - Software interrupts/trap instruktioner fx INT
 - Interrupts
 - Exceptions genereret af hardwaren (udenfor CPU'en)
 - Interrupts
 - Interrupts er, i reglen, **uforudsigelige**
 - Afbryder det kørende program og aktiverer en "interrupt service routine" i styresystemet
 - Kan deaktiveres med en særlig instruktion
 - Bruges til I/O uden polling
 - Processkifte kan aktiveres af timerinterrupt
 - Hardware kan signalere fejlsituationer

Exceptions

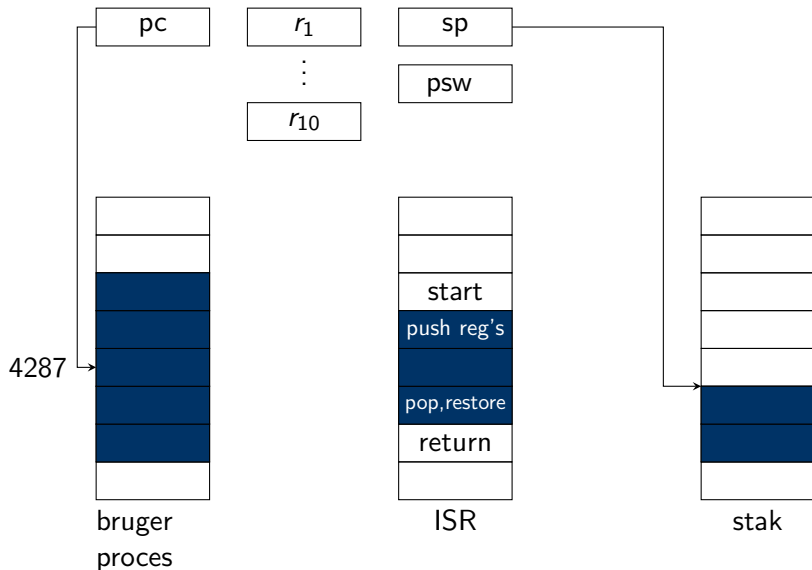
- Medfører kald af en exception handler
 - Programtæller og statusregisteret gemmes på stakken
 - Programtælleren sættes til adressen på exception handleren
 - Kernen skifter til kerne-mode
- Yderligere information gemmes
 - Nuværende processor mode
 - Informationer om hvad der udlæste denne exception
- Exception vektor
 - Tabel med adresser på exception handlerne
 - Initialiseres af styresystemet



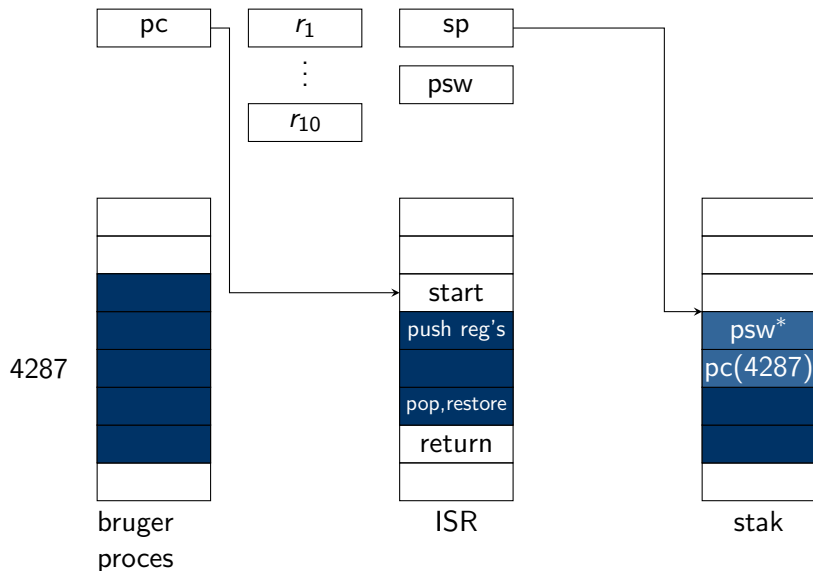
Instruction fetch-løkke med interrupts

- 1 Kopier instruktionen givet ved programtælleren til instruktionsregisteret
- 2 Inkrementer programtælleren
- 3 Afkod instruktionen
- 4 Hent operander
- 5 Udfør instruktionen
- 6 Gem resultatet og opdater statusregisteret
- 7 Hvis interrupts ikke er aktiveret: goto 1
- 8 Hvis der ikke er udestående interrupts: goto 1
- 9 Gem programtæller og statusregister på stakken
- 10 Udlæs interruptnummer: `i`
- 11 Brug interruptvektor til at opdatere programtæller: `ivec[i]`
- 12 Deaktiver interrupts
- 13 Skift til kerne-mode
- 14 goto 1

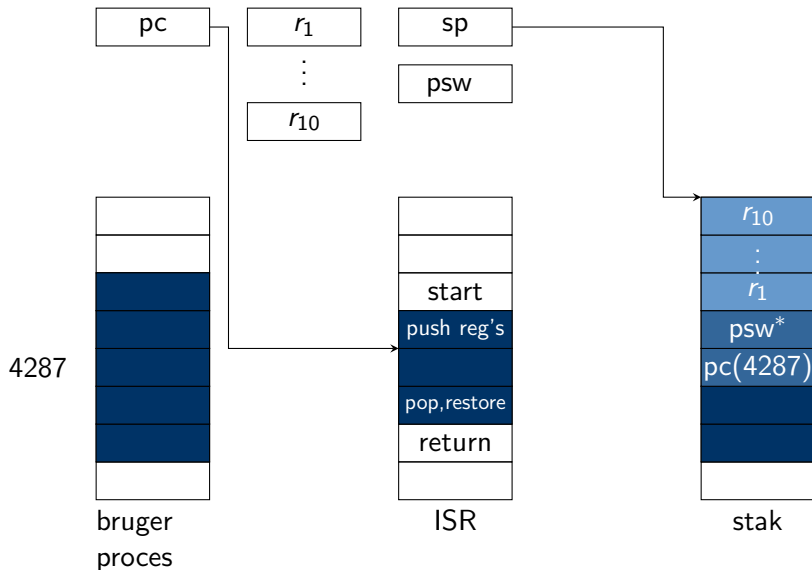
Interrupt Service Routine (ISR)



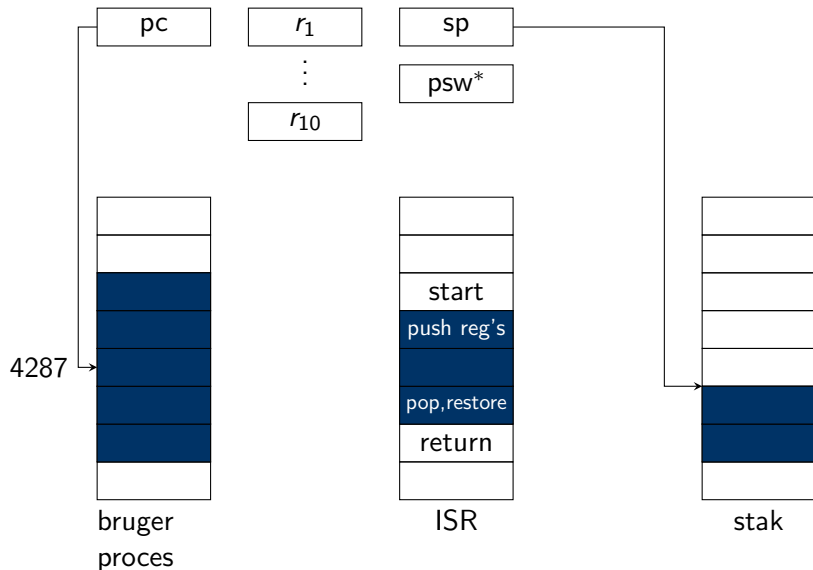
Interrupt Service Routine (ISR)



Interrupt Service Routine (ISR)



Interrupt Service Routine (ISR)



- Interruptdeling
 - Bruges når flere enheder deler samme interrupt
 - Kræver at ISR kan identificere kilden til interruptet
 - **Chain of responsibility** pattern
- Prioriterede interrupts
 - ISR kan afbrydes af interrupts med højere prioritet

Kommunikation med eksterne enheder

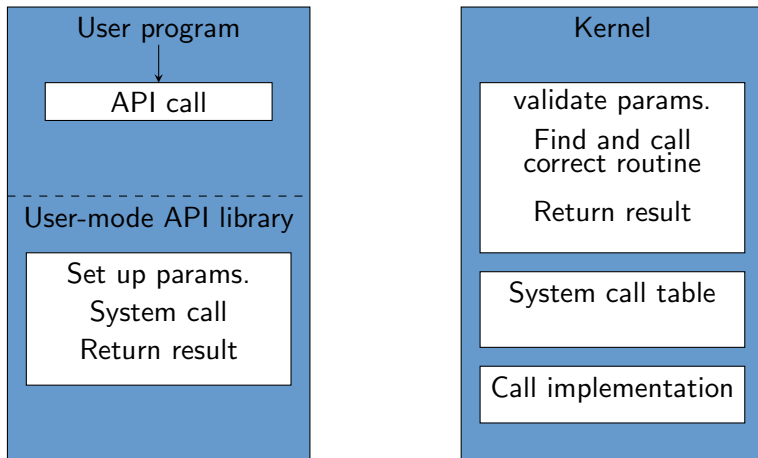
- Eget I/O adresserum
 - Kræver specielle instruktioner til læsning og skrivning
 - Port baseret I/O
- Delt adresserum
 - Afbildet i samme adresserum som hukommelsen
 - Kræver ingen specielle instruktioner
 - Memory-mapped I/O
 - Adresser for primærlager
 - Adresser for I/O
 - Ubrugte adresser

- Programmeret I/O
 - Dataregisteret benyttes til udveksling af data
 - Status- eller kontrolregisteret benyttes til at synkronisere
- Interruptstyret I/O
 - Enheden genererer interrupt når data er klar
 - ISR ansvarlig for at flytte data ud eller ind i dataregisteret
 - Polling kan undgås
 - Processoren er stadig ansvarlig for kopiering af data
- Direct Memory Access (DMA)
 - I/O uddelegeres til en DMA-kontroller
 - Har adgang til adresse- og databussen
 - Interrupt udløses når I/O operationen er fuldført
 - Aflaster processoren

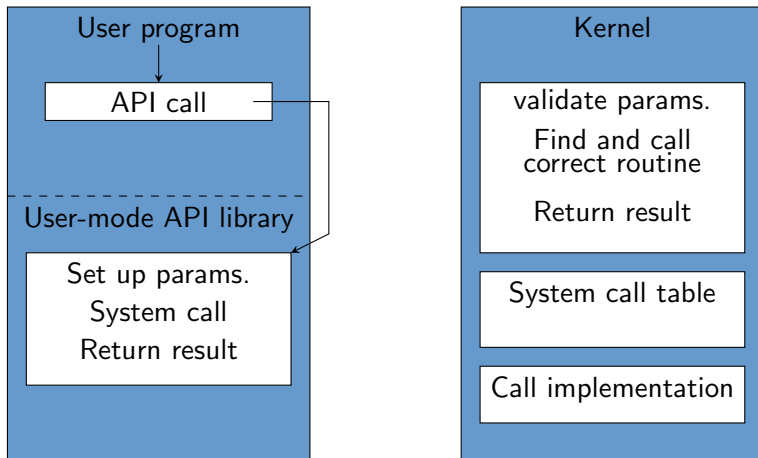
- Hvordan kan processer interagere med styresystemet når de
 - ikke kan skrive til styresystemets hukommelse
 - ikke kan udføre kald til styresystemets kode
 - ikke kan skifte processoren til kerne-mode
- **Exceptions** er den eneste måde at skifte fra user-mode til kerne-mode
- Processoren har instruktioner til at udløse **traps**
 - Kaldes også “software interrupts”
 - Kan bruges til at skifte til kerne-mode og aktivere styresystemet
 - Eksempel: INT instruktionen på Pentium

- Systemkald implementeres ved at udløse en trap
- Nogle arkitekturer har en specialiseret instruktion til systemkald
- Systemkald er nummererede
- CPU register bruges til at udpege den ønskede funktion
- Parametre overføres via registre enten direkte eller indirekte som en adresse der peger på en parameterblok
- Kernen skal validere alle parametre
- Pointere: skal pege på hukommelse der må tilgås af processen

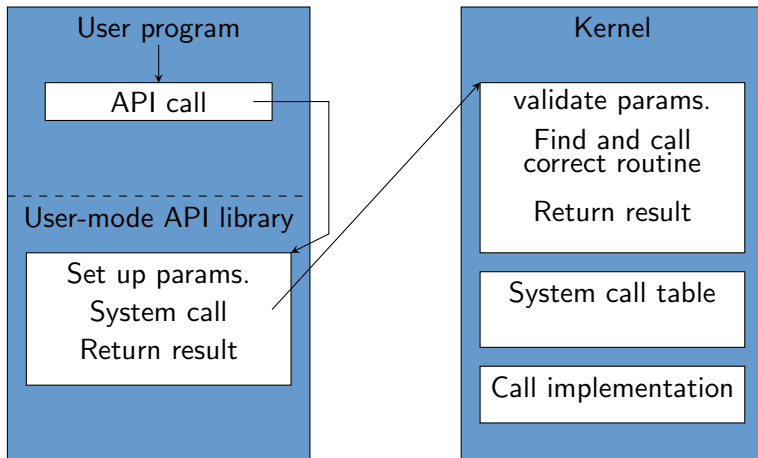
Systemkald



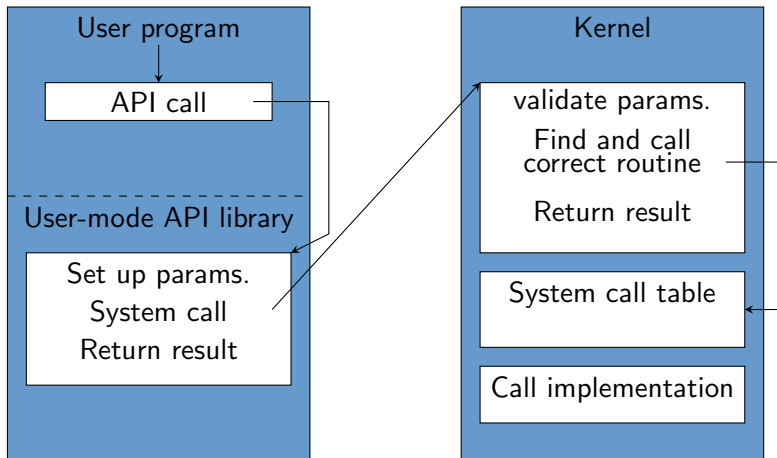
Systemkald



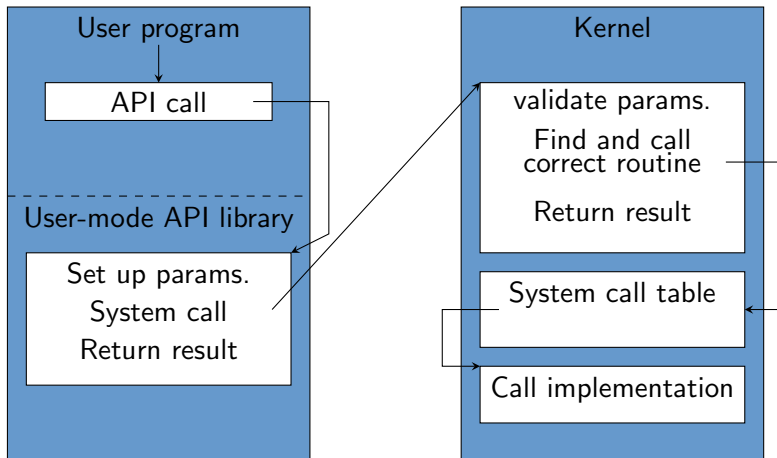
Systemkald



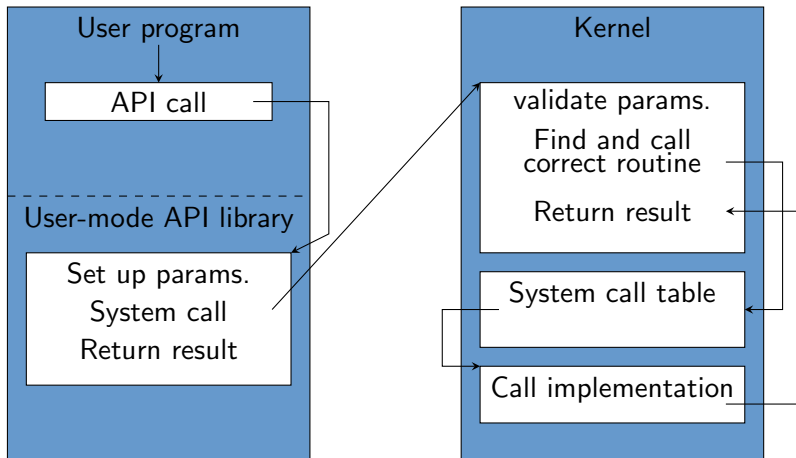
Systemkald



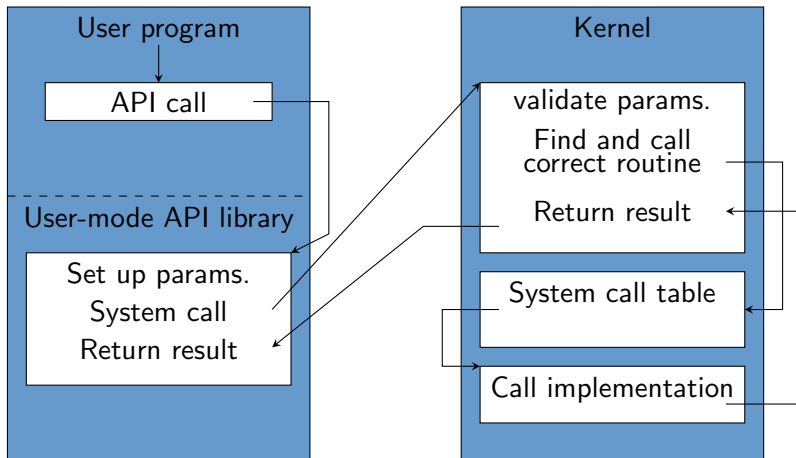
Systemkald



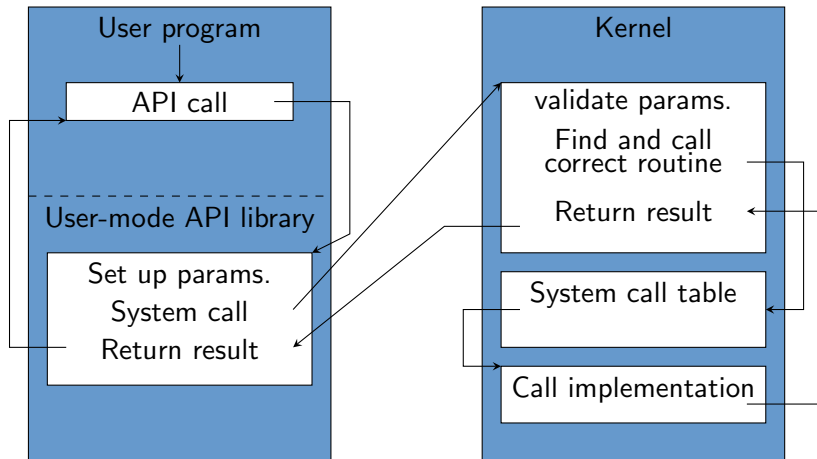
Systemkald

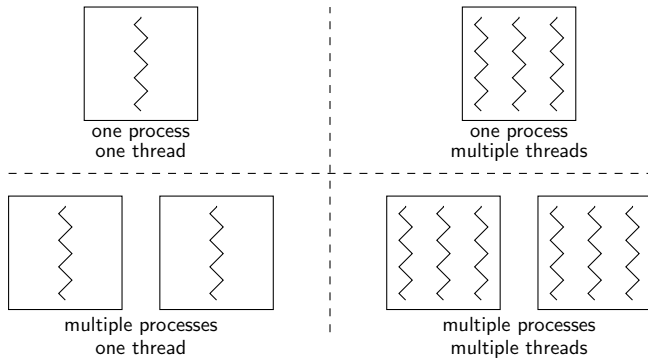


Systemkald

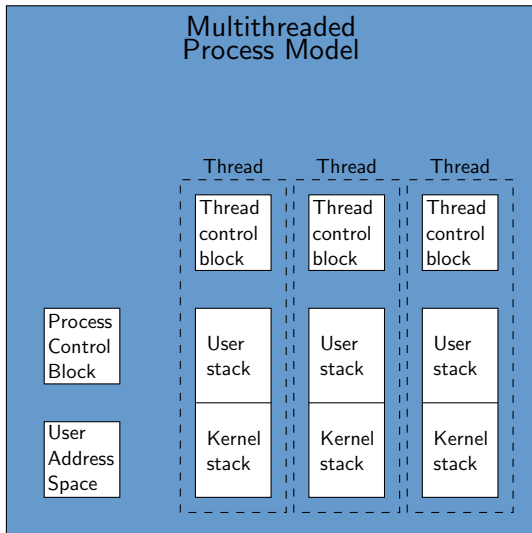
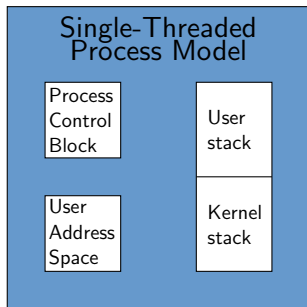


Systemkald

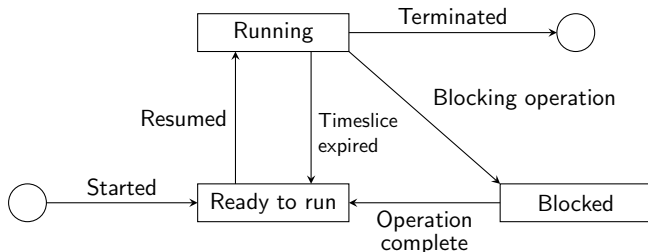




Proces- og trådkontrolblok



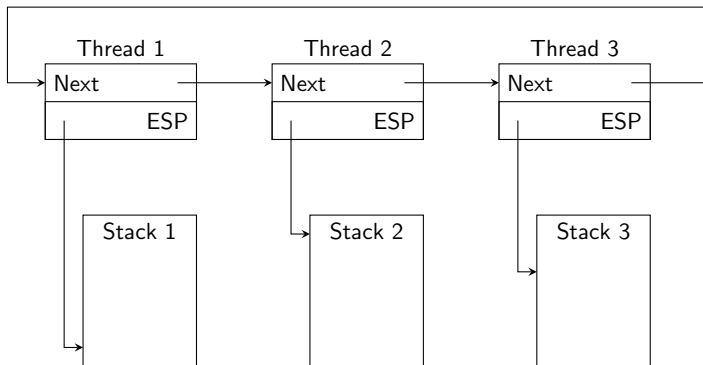
Tilstandsmodellen



- En tråd er i en af flere mulige tilstande
- Klassisk model: fem-tilstandsmodellen

- Ved kontekstskift skal konteksten gemmes
 - Programtæller
 - Statusregister
 - Stakpointer
 - Processorregistre
- Konteksten gemme typisk i trådkontrolblokken

Trådkontrolblok



SUSPEND:

```
PUSHAD  
PUSHFD  
MOV EBX, [CURRENT]  
MOV 4[EBX], ESP  
MOV EBX, 0[EBX]  
MOV [CURRENT], EBX  
MOV ESP, 4[EBX]  
POPFD  
POPAD
```

Trådkontrolblok med stakpointer ved offset 4 og en pointer til næste blok ved offset 0.

Trådoprettele og nedlæggelse

- Oprettelse
 - Alloker ny trådkontrolblok
 - Alloker ny stak og initialiser stakpointer i kontrolblokken
 - Initialiser stak eller kontrolblok med initiale registerværdier
 - Returadressen skal være den første instruktion i den nye tråd
- Nedlæggelse
 - Kald exit funktion
 - Denne nedlægger stakken og kontrolblokken
 - Aktiver næste tråd

Implementation af tråde

- User-space
 - Simpel og portabel
 - Meget hurtig da traps til kernen undgås
 - I/O blokerer alle processens tråde
 - Multiprocessering ikke mulig
 - Begrænset kontrol over scheduling
- Kernel-space
 - Multiprocessering
 - Ingen problemer med blokerende kald
 - Fuld kontrol over scheduling
 - Langsommere p.gr.a. skift til kernel-space (faktor 10)
 - Begrænset skalerbarhed da kerne-lager er begrænset
- Kombineret user- og kernel-space
 - Skalerbarhed, god hastighed, multiprocessering etc.
 - Komplekst
 - Begrænset scheduling, da to schedulere er involveret

Cooperative vs. preemptive

- Cooperative multiprogramming
 - Frivillig tidsdeling
 - Tråd kalder `yield/suspend` for frivilligt at slippe CPU'en
 - Trådskifte kan af og til laves som bivirkning til systemkald
 - Mulighed for at monopolisere processoren
- Preemptive multiprogramming
 - Tvungen tidsdeling
 - Timer bruges til at gennemtvinge trådskifte
- Kombinationer
 - Preemptive multiprogramming med mulighed for at afgive CPU'en før tid
 - Nyttigt, men `yield`-kald skal bruges med stor forsigtighed og kun af eksperter

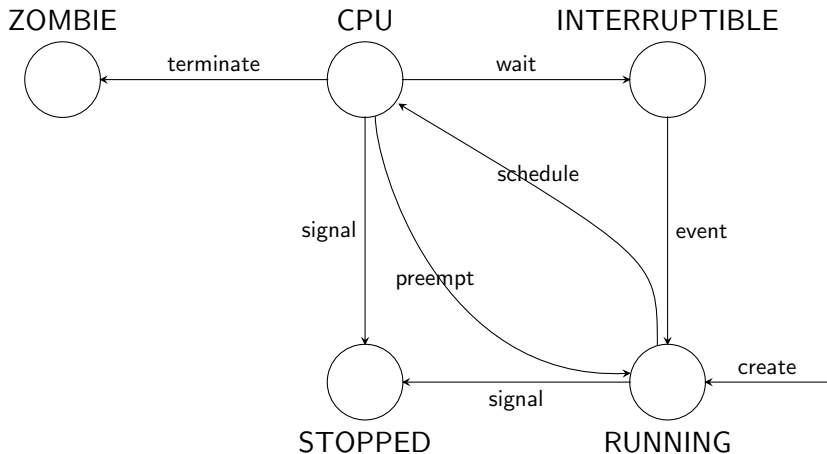
- **Procesbilledet** består af
 - Programteksten
 - Programdata
 - Proces-kontrolblok
 - Stak (en per tråd)
 - Tråd-kontrolblok (en per tråd)
- Proceskontrolblokken indeholder
 - Procesidentifikation
 - Sidetabeller og anden information til hukommelsesstyring
 - File descriptor tabellen
 - Semaforer
 - Ejerskabsinformation
 - Scheduling-information
 - Køretidsinformation (accounting)
 - Børn og forældre
 - etc.

- Tråde er den størrelse, der udføres af CPU'en
- Ved trådkifte skal det rette procesmiljøet genskabes
 - Korrekt proceskontrolblok skal markeres som aktiv
 - Processens sidetabel skal aktiveres
 - TLB skal tømmes

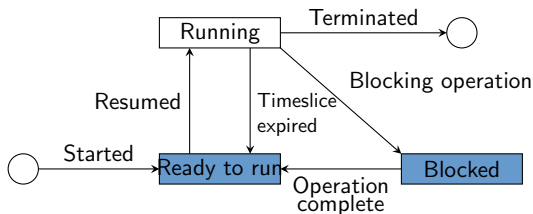
Processer og tråde i Linux

- Minimal forskel på realiseringen af processer og tråde
- Kun kerne-niveau tråde
- Ingen trådkontrolblok
 - Hver tråd har sin egen proceskontrolblok
 - Tråde i samme proces deler dog store dele af proceskontrolblokken (såsom memory management)
 - Tråde tilhører en trådgruppe med eget ID — svarende til proces ID
- Kun muligt fordi procesoprettelse og -skifte er hurtigere i Linux end i de fleste andre styresystemer

Tilstandsmodel for Linux



- Styresystemet schedulerer tråde
 - Men politikken kan afhænge af processerne
- Udfordringer
 - Hvilken tråd skal vælges?
 - Hvor lang tid skal sættes (quantum)?
 - Interaktivitet
 - CPU cache
 - CPU affinity
 - Responstid (realtidskrav)
 - Overhead i beregningstid
 - CPU bound
 - I/O bound



Mange køer at holde styr på

- Ready-køen indeholder tråde der er klar til at blive kørt. Afhængig af scheduleringsprincip kan der være flere ready-køer
- Blokerede tråde ligger i en anden kø (typisk en kø per event type da lineær søgning i en lang kø er for langsom)
- Eksempel: semaforer har en kø af tråde der blokerer (venter) på semaforen
- Timer-kø: kan bruges til at schedulere en tråd til et specifikt tidspunkt; implementeres som difference-list.

- Round-robin
 - Enkelt kø af tråde (run-queue). Første tråd kører i et tidsrum og flytter derefter om bag i køen
 - Kan udvides til multiple køer for flere **prioriteter**
 - Dynamisk prioritering mulig ved at flytte tråde mellem køerne
- Lottery-scheduling
 - Scheduleren trækker lod
 - Prioritering opnås ved at uddele forskelligt antal lodsedler til forskellige tråde
 - Tråde kan handle med lodsedlerne
 - Mulighed for fair fordeling af ressourcer mellem brugere
- Dynamisk prioritering
 - Alle tråde tildeles et “quantum” i forhold til deres statiske prioritet
 - Kørende tråde bruger af deres quantum
 - Tråd med mest tilbage på “kontoen” vælges
 - Når alle tråde i reay-tilstanden har brugt deres quantum gentages algoritmen

Opsummering og næste gang

- Exceptions
- Interrupts
- I/O
- Systemkald
- Processer og tråde: tilstandmodel, oprettelse og nedlæggelse, implementering
- Processer og tråde i Linux
- Scheduling-politikker
- Næste gang: Drivere