

Käyttöjärjestelmät I

Luento 11: SÄIKEET

Stallings, Luku 4.1

KJ-I S2005 / Tiina Niklander, kalvot Auvo Häkkinen

11 - 1

Sisältöä

- n **Prosessi vs. säie**
- n **Miksi säikeitä?**
- n **ULT: Käyttäjätason säikeet**
- n **KLT: Säikelden toteutus ytimessä**
- n **Säikeen tilat**

- n **Käyttöjärjestelmän suorittamisesta**

KJ-I S2005 / Tiina Niklander, kalvot Auvo Häkkinen

11 - 2

Käyttöjärjestelmät I

PROSESSI VS SÄIE

KJ-I S2005 / Tiina Niklander, kalvot Auvo Häkkinen

11 - 3

Prosessi perinteisesti

- n **Resurssien omistaja, jolle allokoitu**
 - u virtuaaliosoitteavaruus, eli suoritussympäristö
 - F prosessin kuva (image): PCB, koodi, data, pino
 - u resursseja
 - F muistia, tiedostoja, I/O-laitteita ...
- n **Vuorottajan hallinnoima kokonaisuus - prosessi on ohjelman suoritus koneessa**
 - u suoritus limittäin muiden prosessien kanssa
 - u prosessiin liittyy tila (Running, ...) sekä prioriteetti

Process, task

KJ-I S2005 / Tiina Niklander, kalvot Auvo Häkkinen

11 - 4

Prosessi nykyaikaisesti

- n **Resurssien kirjanpidon yksikkö, omistaja**
 - u virtuaaliosoitteavaruus, jossa prosessin kuva
 - u laitteiden varaus
 - n **Suojauksen yksikkö**
 - u muistinsuojaus
 - u prosessien välinen kommunikointi
 - u tdstot ja niiden pääsyoikeudet
- mutta Vuorottamisen yksikkö = Säie**
- u CPU suorittaa säikeitä, ei prosesseja

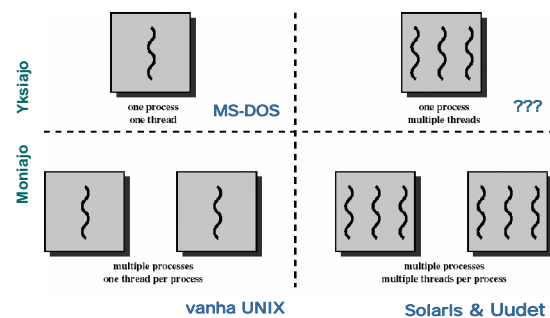
- n **Yksi koodi + resurssit, monta suoritusta**

Thread, Lightweight process

KJ-I S2005 / Tiina Niklander, kalvot Auvo Häkkinen

11 - 5

Prosessi ja säie



KJ-I S2005 / Tiina Niklander, kalvot Auvo Häkkinen

11 - 6

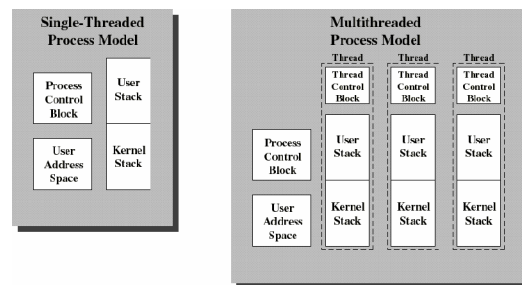
Prosessi voi jakautua säikeisiin

- n ... jos KJ:ssä toteutettu säikeet tai = KLT
- n ... jos käytetään säiekirjastoa = ULT
- n **Säikeellä oma tila (Running, Ready...)**
- n **Säikeellä oma tallealue rekistereille**
 - u mm. omat PC:n ja PSW:n arvot
- n **Säikeellä oma pino**
 - u aliohjelmakutsuja ja paikallisia muuttujia varten
- è **Säikeellä oma kuvaaja**
 - u TCB, Thread Control Block

KJ-I S2005 / Tiina Niklander, kalvot Auvo Häkkinen

11 - 7

Yksi säie vs. Monta säiettä



Säikeen kuvaaja TCB
tallealue rekistereille, prioriteetti, tila, ...

KJ-I S2005 / Tiina Niklander, kalvot Auvo Häkkinen

11 - 8

Yhteiskäyttöiset resurssit

- n **Prosessin säikeet käyttävät yhteistä koodi- ja data-alueita**
 - u viite sivutauluun vain PCB:ssä
 - u mutta jokaisella oma suoritusaikainen pino
- n **Kun säie muuttaa data-alueita (muuttujia), muutos näkyy kaikille prosessin säikeille**
- n **Säikeen avaama tdsto avoinna myös muille prosessin säikeille**
 - u tiedostokuvaajataulu vain PCB:ssä
 - u yhteinen luku/kirjoituspositio?

KJ-I S2005 / Tiina Niklander, kalvot Auvo Häkkinen

11 - 9

Käyttöjärjestelmät I

MIKSI SÄIKEITÄ?

KJ-I S2005 / Tiina Niklander, kalvot Auvo Häkkinen

11 - 10

Miksi säikeitä?

- ⊕ Säikeen luonti (> 10 x) nopeampaa kuin kokonaan uuden prosessin luonti
- **Prosessin säikeiden vuorottaminen nopeampaa kuin prosessien vuorottaminen**
- Ž **Kun säie odottaa, voidaan suorittaa jotain muuta saman prosessin säiettä**
- **Säikeen lopettaminen nopeampaa kuin prosessin lopettaminen**

KJ-I S2005 / Tiina Niklander, kalvot Auvo Häkkinen

11 - 11

Miksi säikeitä?

- **Resurssien jakaminen säikeiden välillä tehokasta**
 - u oletus: yhteiskäyttöisiä
- **Saman prosessin säikeiden välinen kommunikointi helppoa ja nopeaa**
 - u yhteiskäyttöinen data-alue
 - ⊕ kaikilla pääsy globaaleihin muuttujiin
 - u säie hoitaa itse tiedon sovittuun paikkaan, toinen noutaa itse
 - u ei tarvita ytimen apua, ei siirtymisiä etuoik. tilaan
- **Voi helpottaa ohjelmointityötä**

KJ-I S2005 / Tiina Niklander, kalvot Auvo Häkkinen

11 - 12

Miksi säikeitä?

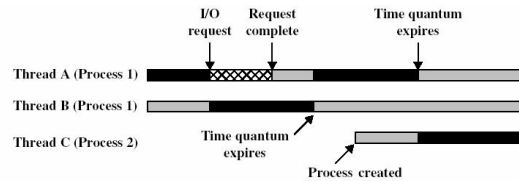
Mutta

- n **synkronointi ja poissulkeminen kokonaan ohjelmoljan vastuulla**
 - u **esim. Yhteisen tietorakenteen muuttaminen**
 - F jos yksi muuttamassa, muut eivät saa käyttää
 - F jos väh. yksi käyttää, kukaan ei saa muuttaa
 - u **esim. Tuottaja ja kuluttaja**
 - F kuluttaja ei saa edetä ennenkuin tuottaja edennyt tiettyyn vaiheeseen

Ö RIO-kurssi

Kuka hyötyy?

Kuva 4.4



Sovellus, jossa selkeästi riippumattomia kokonaisuuksia

- u ts. suoritusjärjestyksellä ei väliä
- u ts. säikeillä ei synkronointitarvetta
- u kukin osa toteutettavissa omana säikeenä

Kuka hyötyy?

Esim: lähiverkon tdstopalvelija

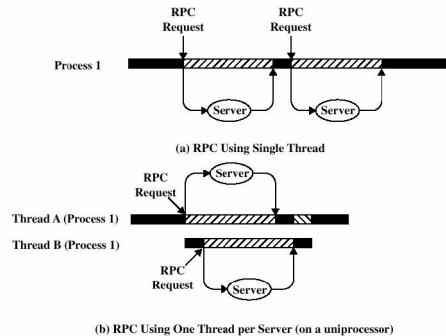
- n Käsiteltävä useita pyyntöjä lyhyessä ajassa
- n Halvempaa luoda/tappaa kutakin pyyntöä kohden oma säle kuin oma prosessi
- n SMP: prosessin säikeet voidaan suorittaa aidosti yhtäaikaan eri prosessoreilla

Esim: komennot valikoista

- n Yksi säle näyttää valikon ja lukee syötteen
- n Toinen säle suorittaa edellistä komentoa
- n Helpompi ohjelmoida

Esimerkki: etäkutsu

Kuva 4.3



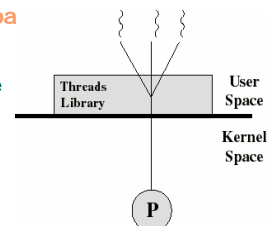
Käyttöjärjestelmät I

KÄYTTÄJÄTASON SÄIKEET

ULT, User Level Threads

Käyttäjätason säikeet

- n Ydin ei tiedä säikeistä
 - u vuorottaa vain prosesseja
- n Sovellus käyttää säikeiden hallintaan sälekirjastoa
 - u kirjanpito, TCB:t
- n Sovellus huolehtii itse säikeidensä vuorottamisen
 - u ohjelmoijan vastuulla
 - u ei käytä ytimen koodia
 - u ei keskeytysohjattua



Säikekirjaston perusrutiinit

Karkea jako:

- n **Säikeen luominen**
 - u kirjanpito säikeestä ja sen tilasta
 - n **Säikeen etenemisen estäminen**
 - u rekistereiden tallettaminen
 - n **Säikeen etenemisen salliminen**
 - u rekistereiden palauttaminen
 - n **Säikeen lopettaminen**
- n Kohdat 2 ja 3 = synkronointi

KJ-I S2005 / Tiina Niklander, kalvot Auvo Häkkinen

11 - 19

POSIX threads (pthreads)

- n pthread-kirjastossa yli 60 funktiota
- n **pthread_create()**
 - u parametrina funktio, josta suoritus alkaa
- n **pthread_exit()**
 - u lopeta säikeen suoritus
- n **pthread_join()**
 - u odota parametrina annetun säikeen loppumista
- n **Synkronointi, poissulkeminen (semaforit)**
 - u pthread_mutex_init() / _destroy()
 - u pthread_mutex_lock() / _trylock() / _unlock()
- n **Ynnä muita funktioita**
 - u sched_yield(): luovu vapaaehtoisesti CPU:sta

KJ-I S2005 / Tiina Niklander, kalvot Auvo Häkkinen

11 - 20

Esimerkki

```
void main() {
    pthread_t thr1, thr2;
    char *msg1 = "Hello";
    char *msg2 = "World";

    pthread_create(&thr1, pthread_attr_default,
                  (void*)&print_message_function, (void*)msg1);
    pthread_create(&thr2, pthread_attr_default,
                  (void*)&print_message_function, (void*)msg2);
    exit(0);
}

void print_message_function(void *ptr){
    char *message;
    message = (char *) ptr;
    printf("%s ", message);
}
```

KJ-I S2005 / Tiina Niklander, kalvot Auvo Häkkinen

11 - 21

Käyttäjätason säikeet

Hyötyjä

- n **Vuorottaminen nopeaa**
 - u ei KJ:n apua
 - u ei keskeytystä
 - u ei prosessinvaihtoa!
- n **Ohjelmoija voi valita sopivan vuorottamistavan**
 - u KJ ei tunne sovelluksen tarpeita
- n **Sovellus siirrettävissä helposti ympäristöihin, joissa sama säikekirjasto**

Haittoja

- n **Kun säikeen palvelupyyntö aiheuttaa odotusta, kalkki muut saman prosessin säikeet odottavat**
- n **Saman prosessin säikeet eivät voi olla suorituksessa usealla prosessorilla yhtäaikaan**
 - u Ydin vuorottaa vain prosesseja!

KJ-I S2005 / Tiina Niklander, kalvot Auvo Häkkinen

11 - 22

Käyttöjärjestelmät I

YTIMEN SÄIKEET

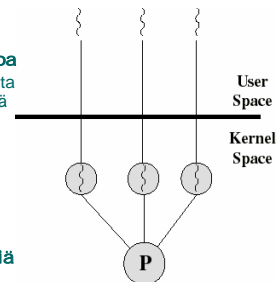
KLT, Kernel Level Threads

KJ-I S2005 / Tiina Niklander, kalvot Auvo Häkkinen

11 - 23

Säikeiden toteutus ytimessä

- n **Säikeiden hallinta kokonaan KJ:n ytimessä**
- n **Toteutus ei käytä kirjastoa**
 - u ohjelmoijalle näkyvä rajapinta voi olla silti sama kuin edellä
 - u palvelupyynnöt
- n **Ydin tietää säikeistä**
 - u prosesseista PCB:t
 - u niiden säikeistä TCB:t
- n **Vuorottaminen KJ:n helinä**
 - u aikaviipale säikeelle



KJ-I S2005 / Tiina Niklander, kalvot Auvo Häkkinen

11 - 24

Säikeiden toteutus ytimessä

Hyötyjä

- Prosessin säikeitä voi olla yhtäaikaa suorituksessa eri prosessoreilla
- Jos säle Blocked- tilaan, muut prosessin säikeet voivat silti jatkaa
- Myös KJ-ytimen toteutus voi käyttää säikeitä

Haittoja

- Säikeen vaihto vaatii aina 2 vaihetta:
 - keskeytys + siirtyminen KJ:hin (-> etuoik. tila)
 - vuorottaminen (->kjätila)
- Vaihto hitaampaa kuin kirjastoa käytettäessä
 - ks. taulukko 4.1

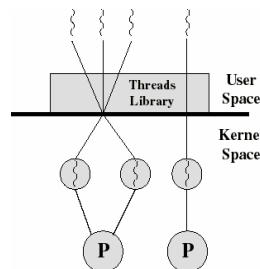
Operation	User-Level Threads	Kernel-Level Threads	Processes
Null Fork	34	948	11,300
Signal Wait	37	441	1,840

Käyttöjärjestelmät I

KULTAINEN KESKITIE ?

Solaris

- Yhdistää molempien parhaat piirteet
- Luontil käyttäjätilassa
- Synkronointi käyttäjätilassa
- Pääosa vuorottamisesta käyttäjätilassa
- Ohjelmoija voi *ilittää* käyttäjätason säikeet ytimen säikeiksi haluamallaan tavalla



Solaris

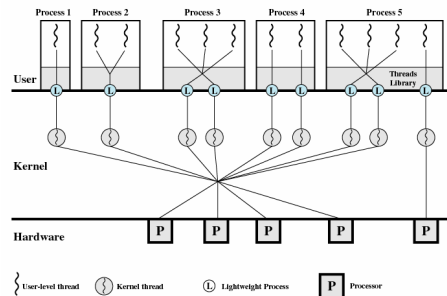


Figure 4.15 Solaris Multithreaded Architecture Example

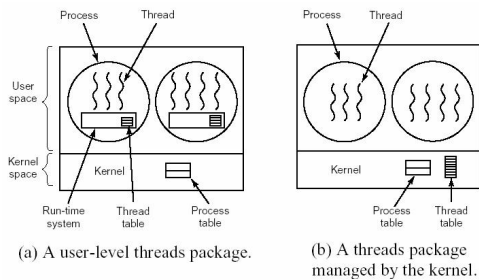
Käyttöjärjestelmät I

SÄIKEEN TILAT

Säikeen tilat

- Perustilat: Running, Ready, Blocked**
 - kuten prosessilla
- Suspend-tila koskee aina koko prosessia**
 - liittyy heittovaihtoon
 - osoiteavaruus prosessitason käsite
 - mm. koodialue + globaali data yhteiskäytössä
- Kun prosessi joutuu Suspend-tilaan, joutuvat kaikki sen säikeet odottamaan**
- Kun prosessi lopetetaan, poistetaan samalla kaikki sen säikeet**

ULT-säie vs KLT-säie



Tan01 kuva 2-13

KJ-I S2005 / Tiina Niklander, kalvot Auvo Häkkinen

11 - 31

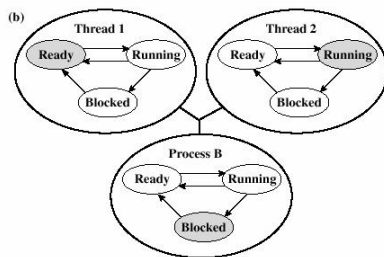
ULT-säikeen vs Prosessin tila

- n **KJ:n ydin ei tiedä ULT-säikeistä**
 - u KJ vuorottaa prosessitasolla
 - F Ready-jonossa prosesseja
 - u aikaviipale prosessille
 - u säiekirjasto huolehtii säikeiden vuorottamisesta prosessin aikaviipaleen sisällä
- n **Jos ULT-säie odottaa palvelupyynnössä, prosessi joutuu Blocked-tilaan**
- n **mutta säiekirjaston kirjanpidossa säie edelleen Running-tilassa**
- Ø ULT-säikeen ja prosessin tila erillisiä

KJ-I S2005 / Tiina Niklander, kalvot Auvo Häkkinen

11 - 32

ULT-säikeen vs Prosessin tila



Kuva 4.7

KJ-I S2005 / Tiina Niklander, kalvot Auvo Häkkinen

11 - 33

KLT-säikeen vs Prosessin tila

- n **KJ tietää säikeiden olemassaolosta**
 - u KJ ylläpitää säiekuvaajia
 - u KJ vuorottaa säietasolla
 - F jos prosessin aikaviipaleta jäljellä, vuorota saman prosessin säikeitä tai
 - F koko aikaviipale säikeelle
- n **Jos KLT-säie odottaa palvelupyynnössä, prosessin muut säikeet voivat silti jatkaa**

KJ-I S2005 / Tiina Niklander, kalvot Auvo Häkkinen

11 - 34

Mahdollisia tilan vaihtoja

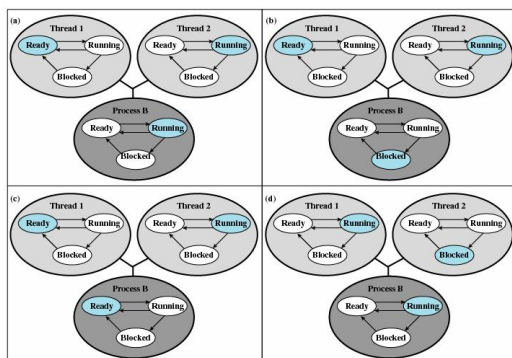


Figure 4.7 Examples of the Relationships Between User-Level Thread States and Process States

Kertauskysymyksiä

- n Miten prosessi ja säie eroavat toisistaan?
- n Mitä yhteistä niillä on?
- n Mitä hyötyä säikeistä?
- n Miten ULT ja KLT eroavat toisistaan?
- n Miksi ULT säikeiden vuorottaminen nopeampaa kuin KLT säikeiden?
- n Miksi ULT säikeen Blocked-tila vie prosessin Blocked-tilaan, mutta KLT säikeen ei?

KJ-I S2005 / Tiina Niklander, kalvot Auvo Häkkinen

11 - 36

Käyttöjärjestelmät I

KJ:N SUORITTAMISESTA

(jäi käsittelemättä luennolla 3 aikapulan vuoksi)

KJ-I S2005 / Tiina Niklander, kalvot Auvo Häkkinen

11 - 37

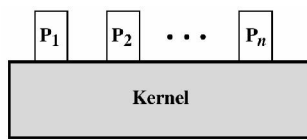
KJ:n suorittamisesta

- n Myös KJ eräs CPU:n suorittamista käskykokoelmista
- n Käyttäjätilassa / etuoikeutetussa tilassa
- n KJ:n osat käsittelevät yhteisiä data-alueita
 - u melkein kaikki käyttävät PCB:tä
- n Onko KJ myös prosessi?

KJ-I S2005 / Tiina Niklander, kalvot Auvo Häkkinen

11 - 38

j KJ etuoikeutetussa tilassa

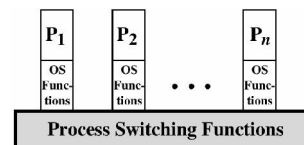


- n **Prosessi vain käyttäjätilan käsite**
 - u KJ:n osat eivät jonota
- n **KJ:llä omat muistialueensa: koodi, data, pino**
- n **KJ:n osat suoritetaan omillaan etuoik. tilassa**
 - u oikeus tehdä kaikkia KJ:n toimintoja kaikissa osissa
- ~ **vanha monoliittinen KJ**

KJ-I S2005 / Tiina Niklander, kalvot Auvo Häkkinen

11 - 39

k KJ prosessin ympäristössä



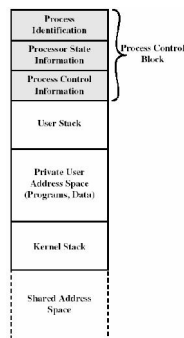
- n **KJ yhteiskäyttöisellä muistialueella**
 - u kaikkien prosessien osoitevaruudessa
- n **Prosessi itse suorittaa KJ:n rutiineja**
 - u hallittu siirtyminen keskeytyksellä, etuoikeutettu tila
- n **Kontrolli prosessilta poissa vain, kun synkronointi tai vuorottaminen vaatii**
- ~ **uudempi monoliittinen KJ**

KJ-I S2005 / Tiina Niklander, kalvot Auvo Häkkinen

11 - 40

KJ prosessin ympäristössä

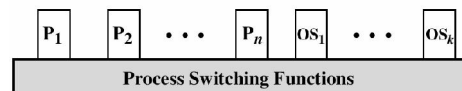
- n **KJ:n koodi ja data yhteisellä muistialueella**
- n **Prosessi käyttää kernel-pinoa, kun suorittaa KJ:n koodia, muulloin normaalla pinoaan**
- n **Prosessi voi odottaa KJ:n koodissa**
- n **Uuselta KJ:n osia voi olla yhtäaikaan kesken eri prosessien ympäristöissä**
 - u suoritukseen vuorottajan kautta



KJ-I S2005 / Tiina Niklander, kalvot Auvo Häkkinen

11 - 41

l KJ = joukko palveluprosesseja



- n **Monet KJ:n palveluista erillisiä prosesseja**
 - u odottavat Blocked/Ready-jonossa
 - u kullakin oma osoitevaruus
 - u tarvittaessa etuoikeutetussa tilassa, erilaisia oikeuksia
- n **Vuorottaminen prosessien ulkopuolella**
- n **Sanomanvälitys: pyyntö-vastaus mekanismi**
 - u palvelupyyntö: lähetä / vastaanota sanoma
 - u sopii myös moniprosessori / hajautettuihin järjestelmiin
- n **Jos ytimessä vain laiterilippuvat toiminnot = mikrokernel**

KJ-I S2005 / Tiina Niklander, kalvot Auvo Häkkinen

11 - 42