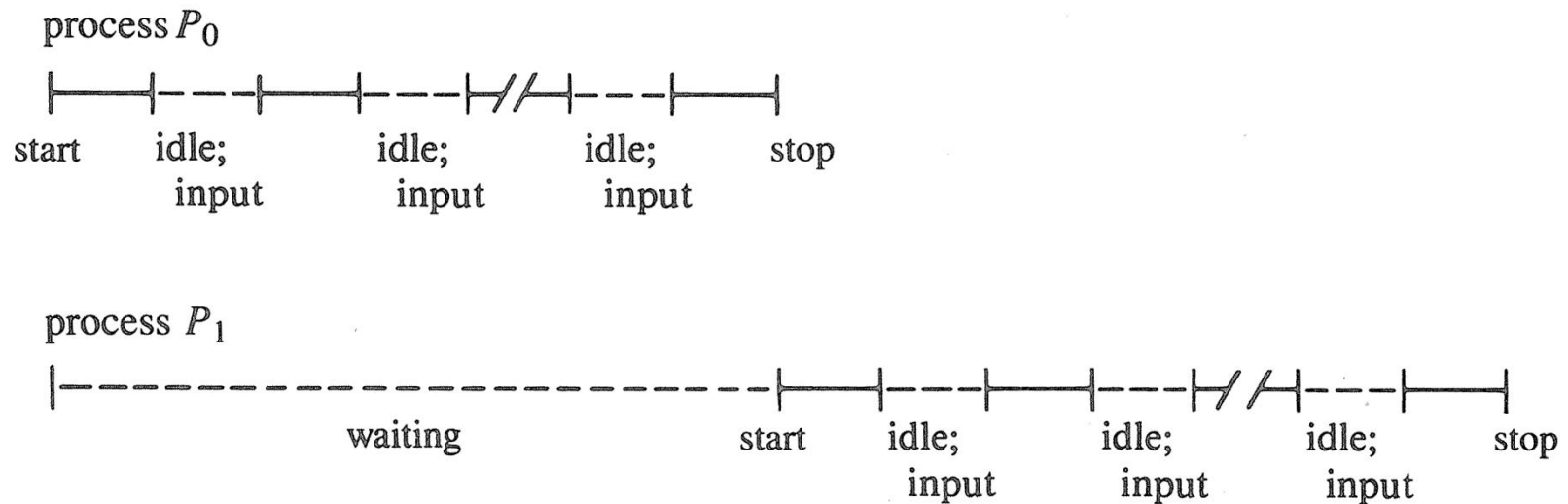


## Palvelut

- Käyttöjärjestelmän tehtävänä on tarjota ohjelmoijalla erilaisia palveluita
- Tyypillisin palvelu, jota sulautetut käyttöjärjestelmät tarjoavat on prosessien hallinta
  - Terminologia on hyvin vaihtelevaa: process, task, thread
- Prosessien hallintaan kuuluu
  - Prosessien käynnistäminen ja päättäminen
  - Suoritettavan prosessin valinta (skeduleri)
- Prosessien hallinnan lisäksi tarvitaan synkronointipalveluita, joilla voidaan ohjata suoritettavien prosessien toimintaa
  - Lukot (mutex)
  - Semaforit
  - Monitorit
  - Jonot
- Lisäksi voi olla muistinhallinta, syöttö ja tulostus, keskeytysrutiinien hallinta, tiedostojärjestelmä, verkkoyhteydet, jne.

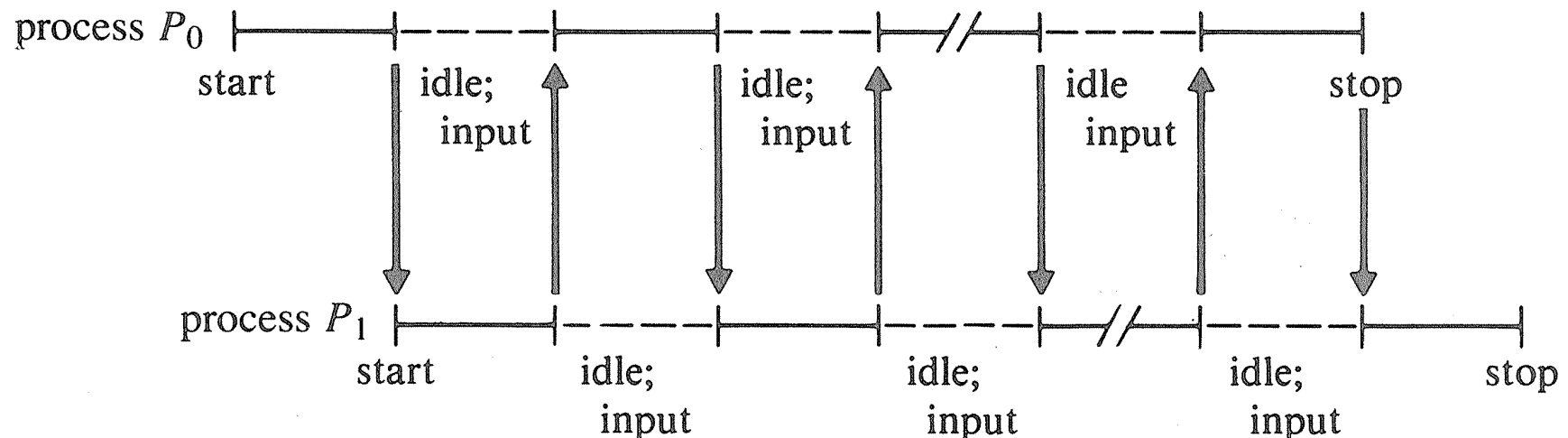
## Moniajo (multitasking)

- Kahden toisistaan riippumattoman prosessin suorittaminen peräkkäin ei ole vasteajan ja suoritusnopeuden kannalta järkevää
- Hyvin monet prosessit käyttävät suurimman osan ajastaan odotteluun



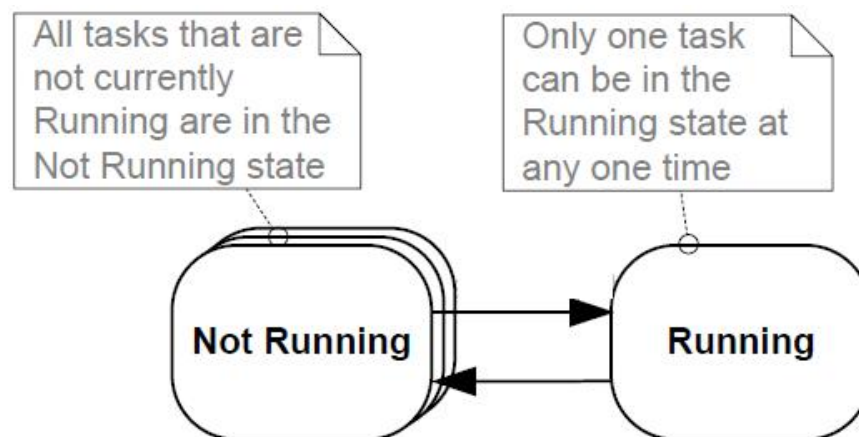
## Moniajo (multitasking)

- Moniajossa skeduleri lomittaa prosessit siten että yhden prosessin odotusaikana jokin toinen prosessi pääsee suoritukseen
  - Kokonaissuoritus aika lyhenee, koska odottamisen määrä pystytään minimoimaan
- Skedulointi
  - Pre-emptive = käyttöjärjestelmä voi keskeyttää prosessin suorituksen milloin tahansa ja siirtää suoritukseen toiselle prosessille
  - Co-operative = prosessin pitää itse luovuttaa suoritukseen



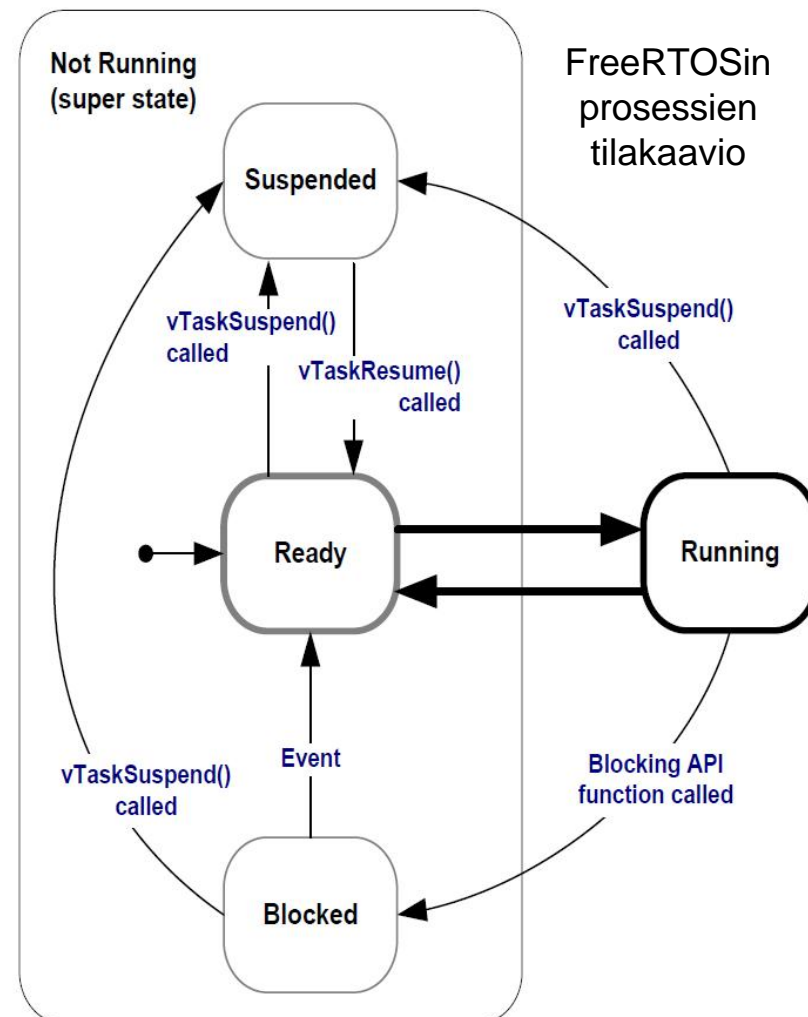
## Moniajo (multitasking)

- Yhdellä prosessorilla voi olla vain yksi prosessi kerrallaan suorituksessa ja muut prosessit ovat ei-suorituksessa
- Vaihtoa prosessista toiseen sanotaan kontekstin vaihdoksi (context switch)
  - Konteksti sisältää kaiken sen prosessorin tilatiedon, joka tarvitaan, jotta prosessi voi jatkaa suoritustaan myöhemmin laskennan häiriintymättä
  - Kontekstin vaihtoon kuluu aina jonkun verran aikaa, joten prosessien määrää ei voi kasvattaa rajattomasti

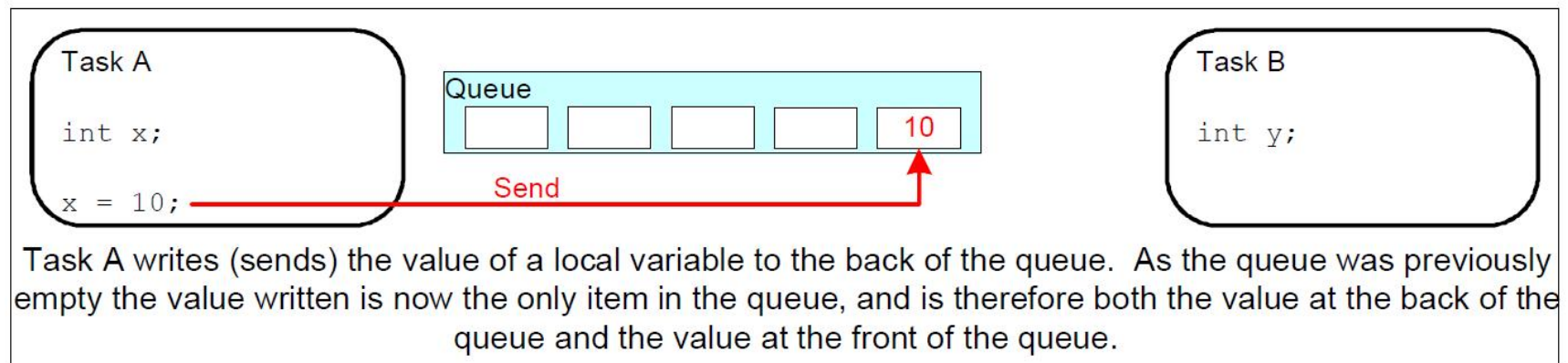
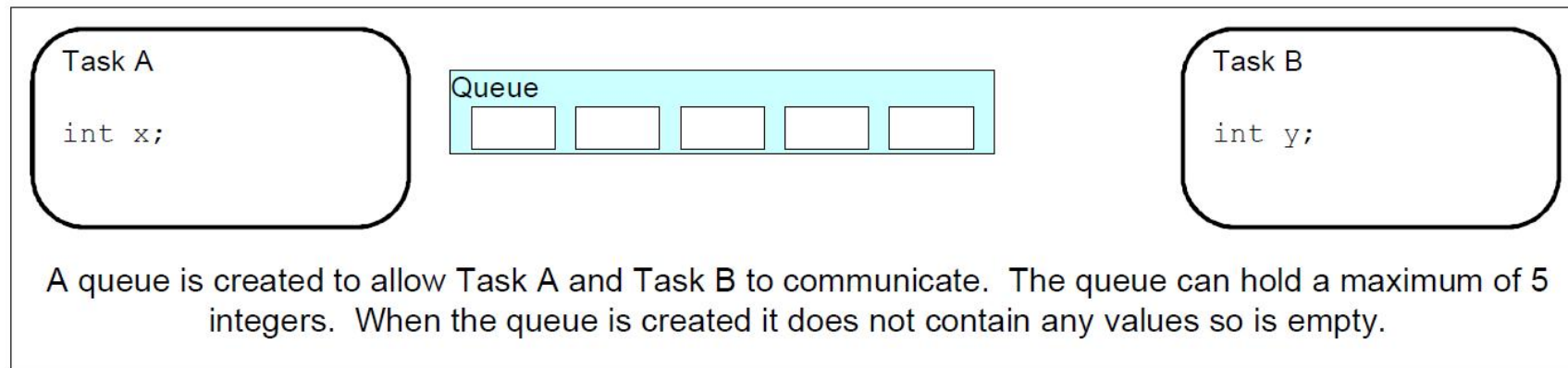


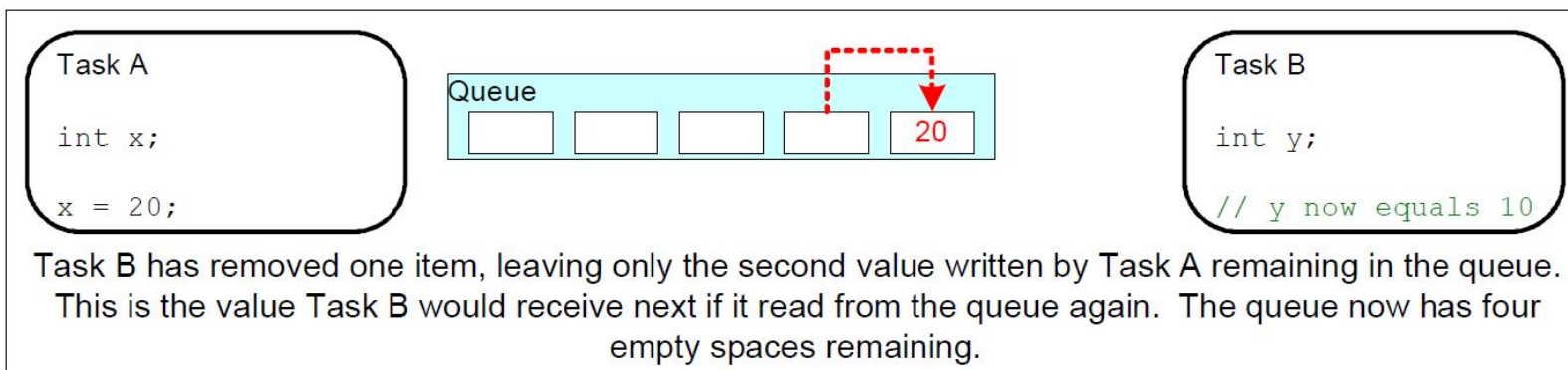
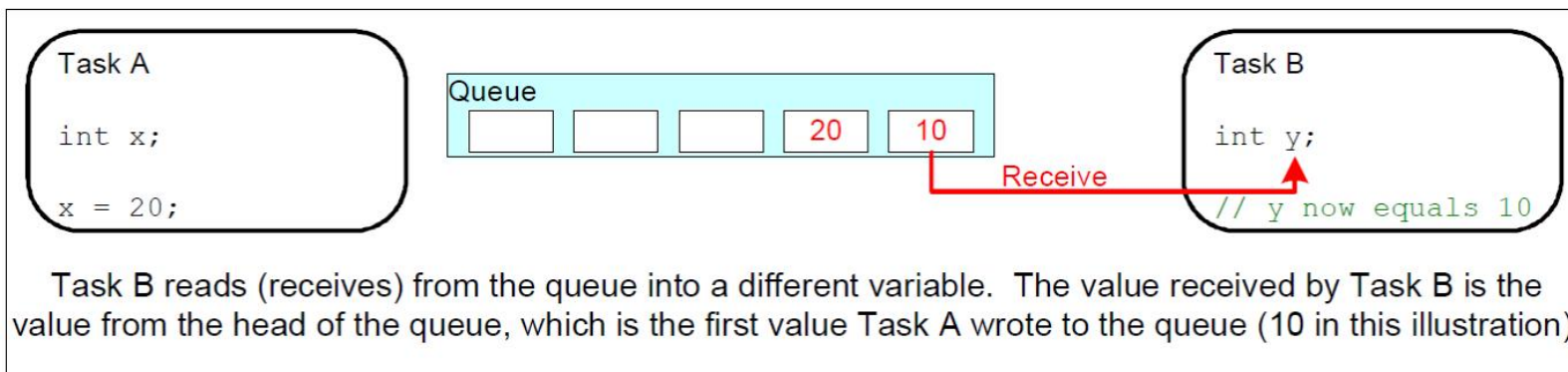
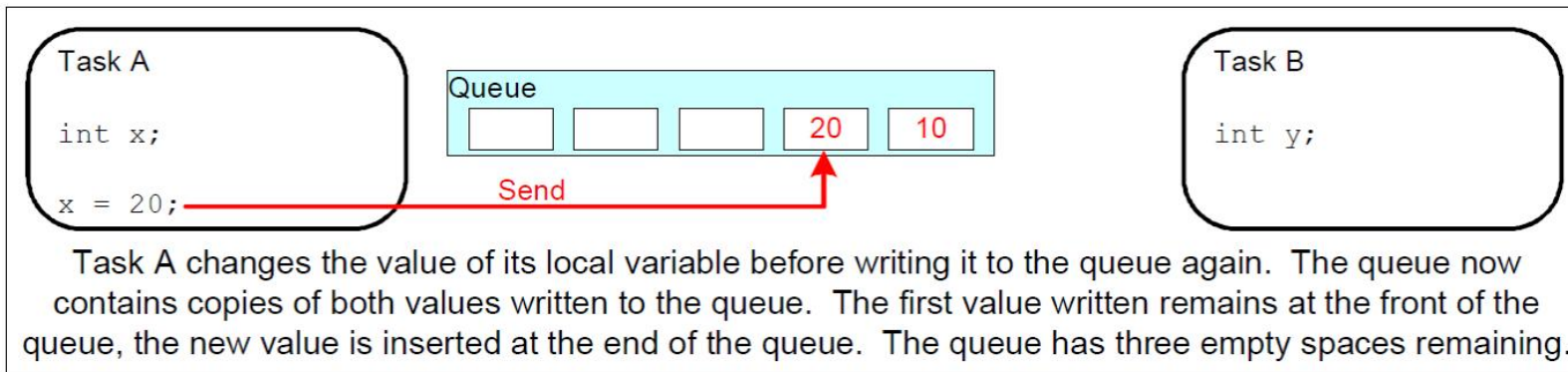
## Moniajo (multitasking)

- Ei ajossa oleva prosessi voi olla useassa eri tilassa
  - Ready (valmis) – odottaa omaa suoritustilaa
  - Blocked – prosessi ei ole valmiina, vaan odottaa esimerkiksi vapautuvaa resurssia tai alkijonosta
  - Suspended – prosessi on pysäytetty ja se pitää palauttaa ajokelpoiseksi käyttöjärjestelmäkutsulla

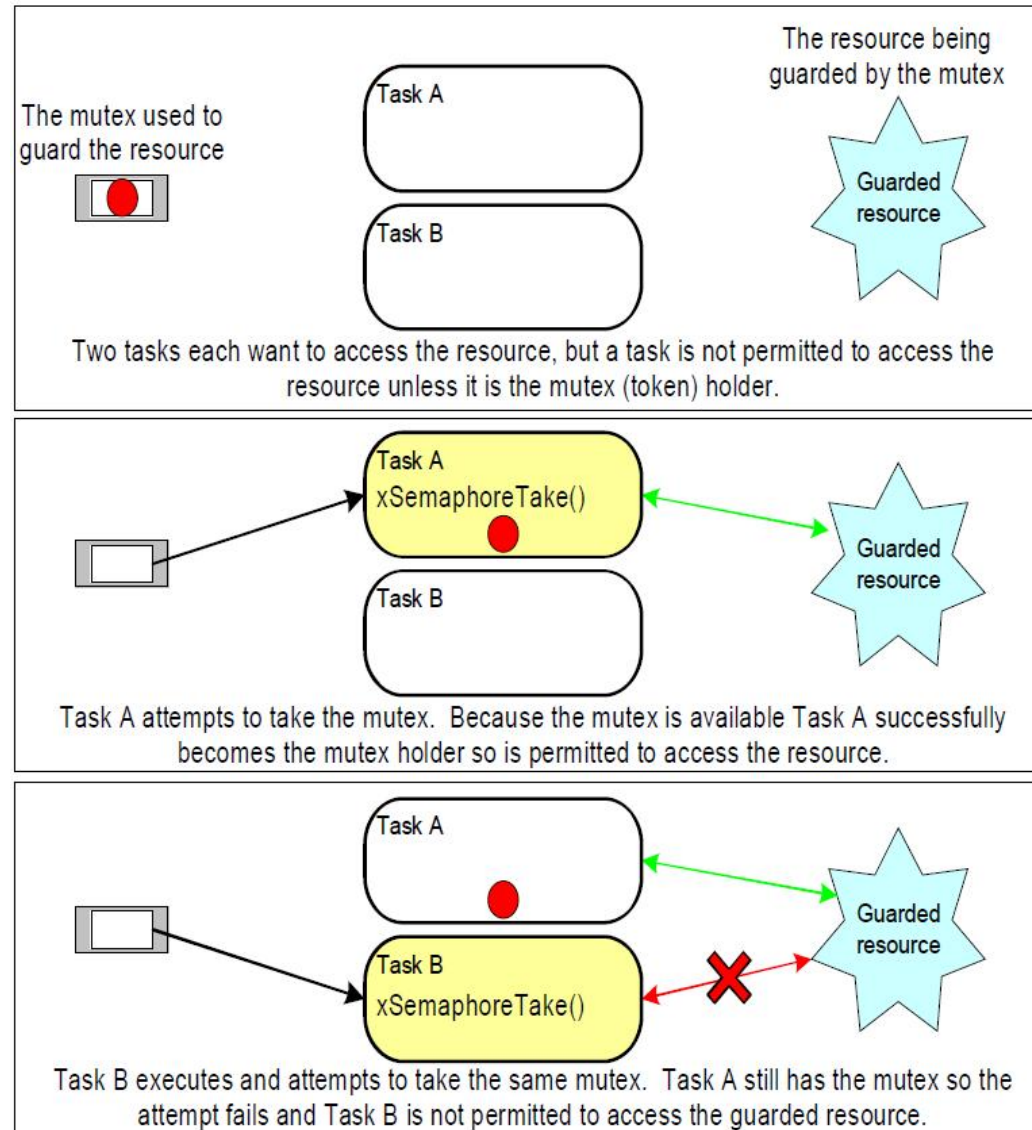


## Jonot

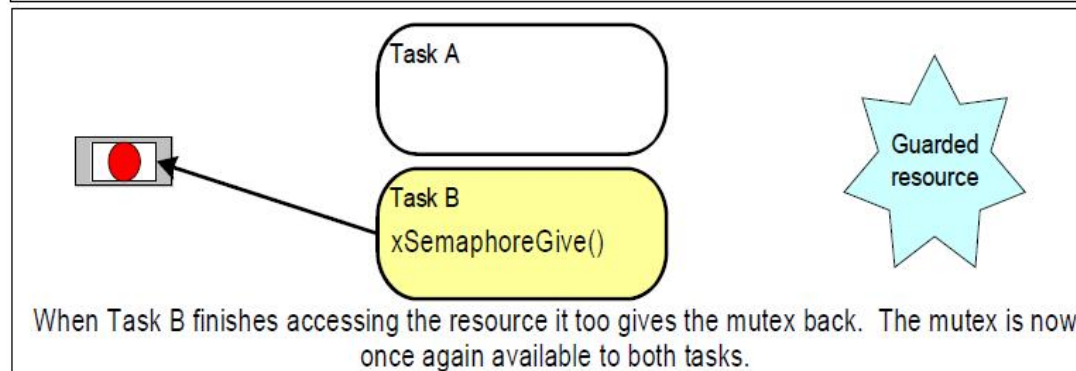
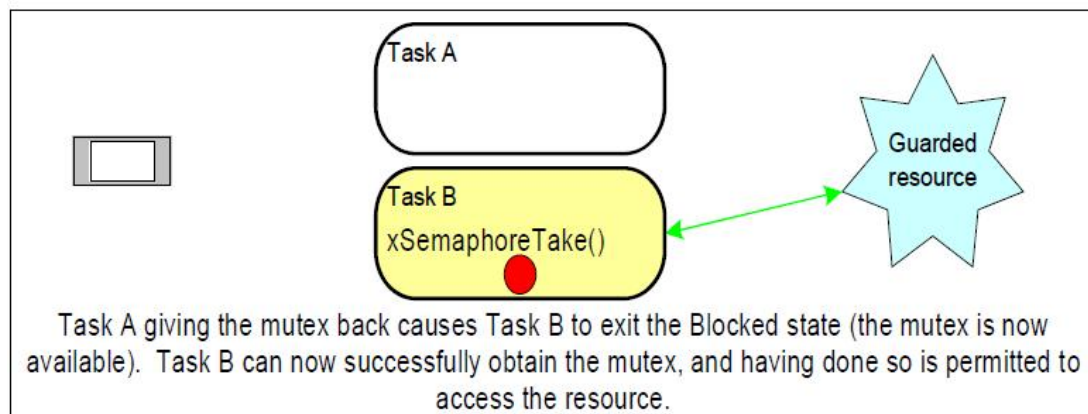
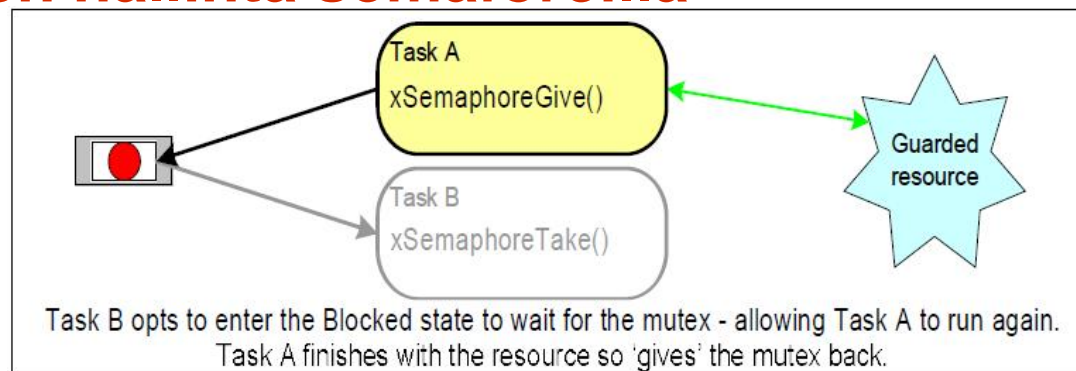




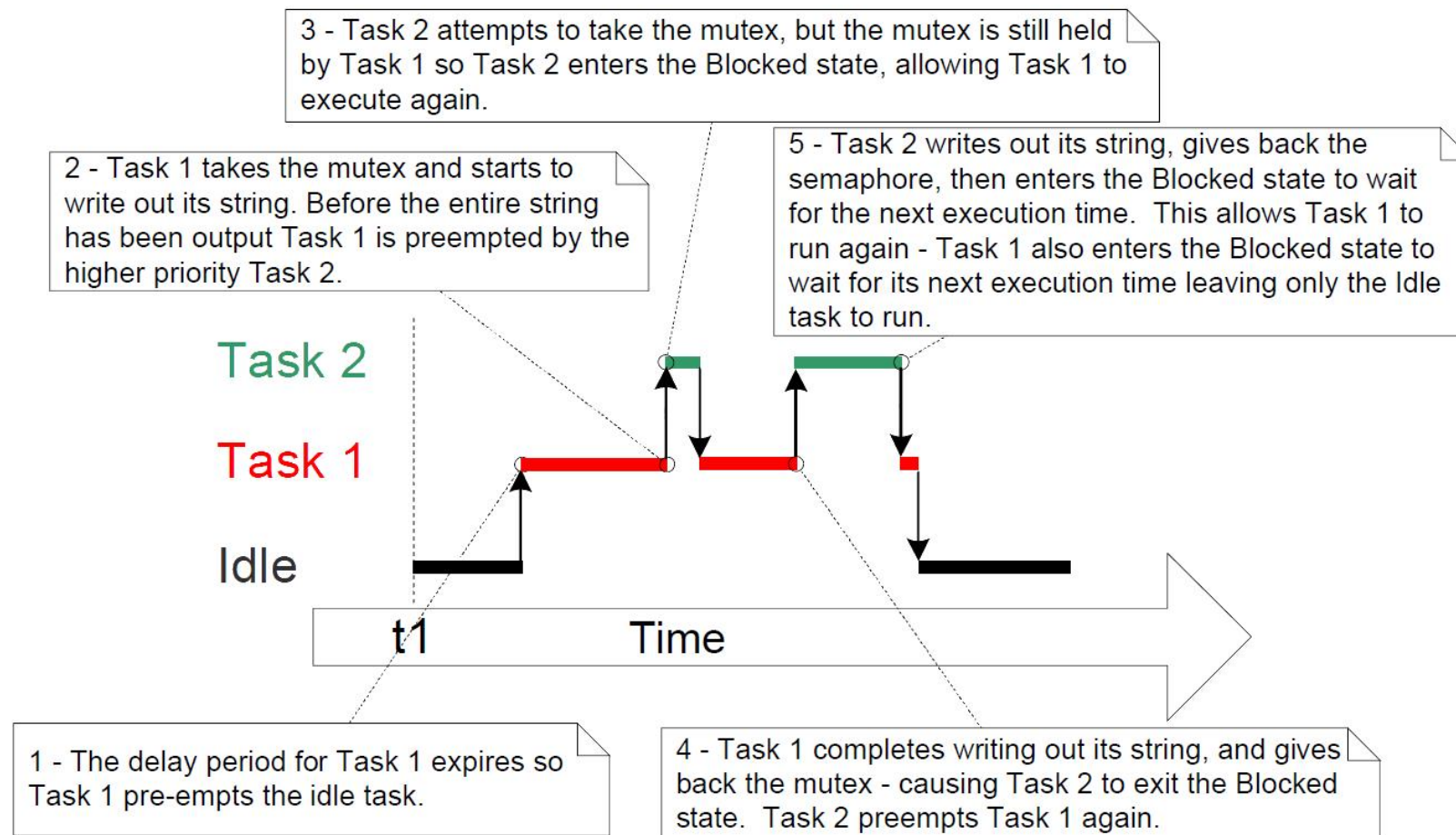
## Resurssien hallinta semaforeilla



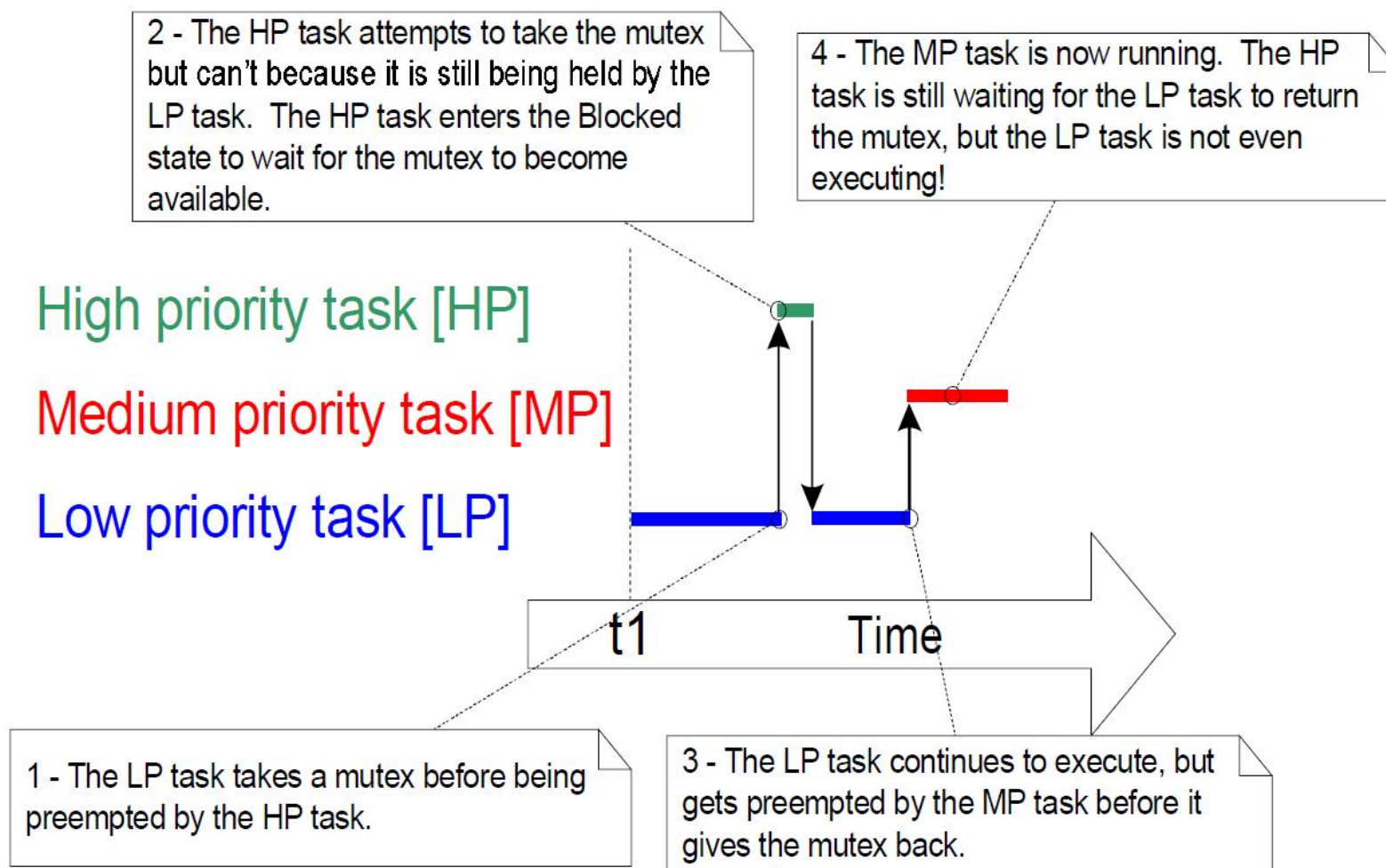
## Resurssien hallinta semaforeilla



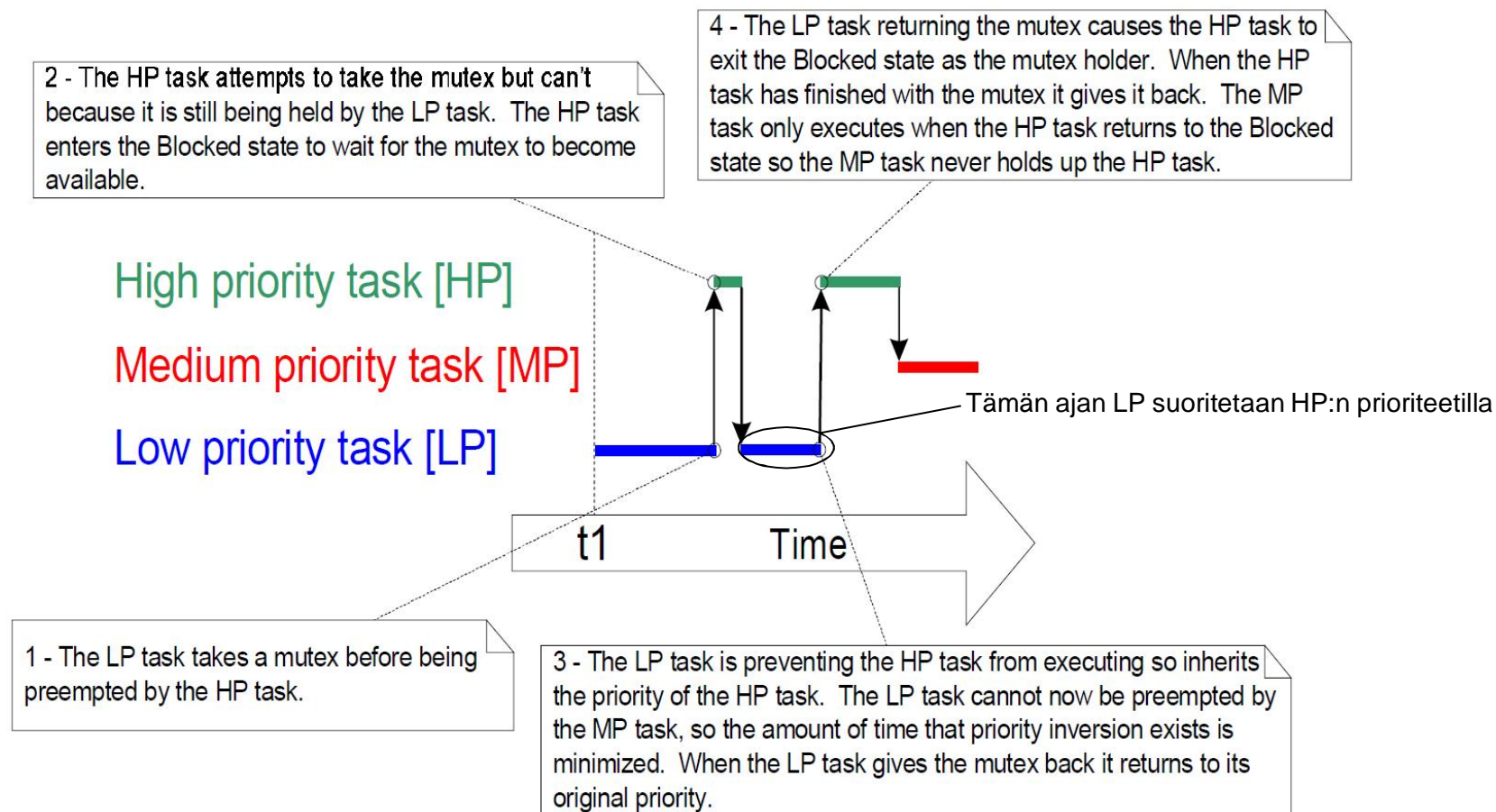
## Käänteisprioriteettiongelma (priority inversion)



## Käänteisprioriteettiongelma (priority inversion)

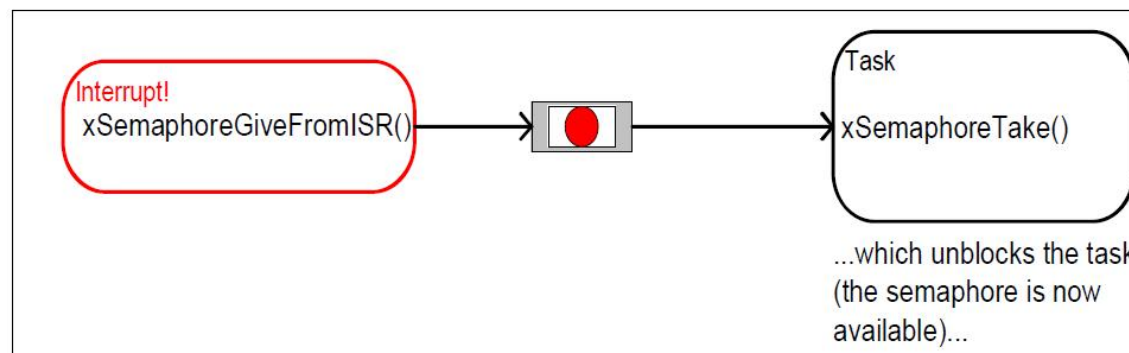
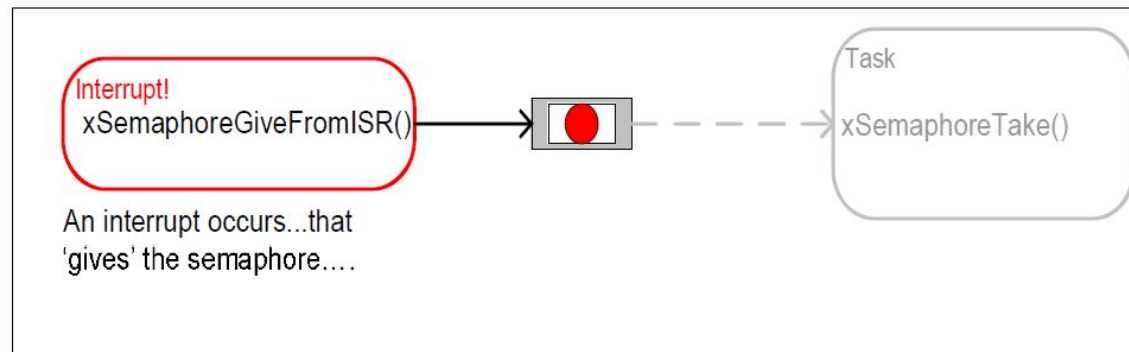


## Prioriteetin perintä (priority inheritance)

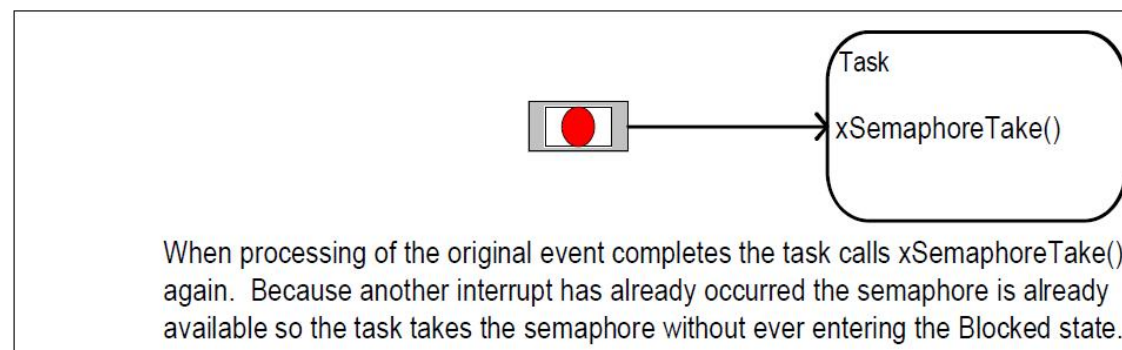
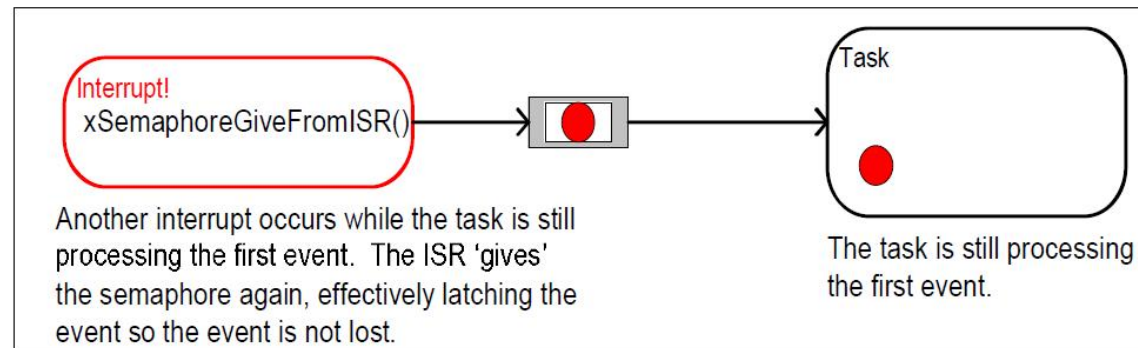
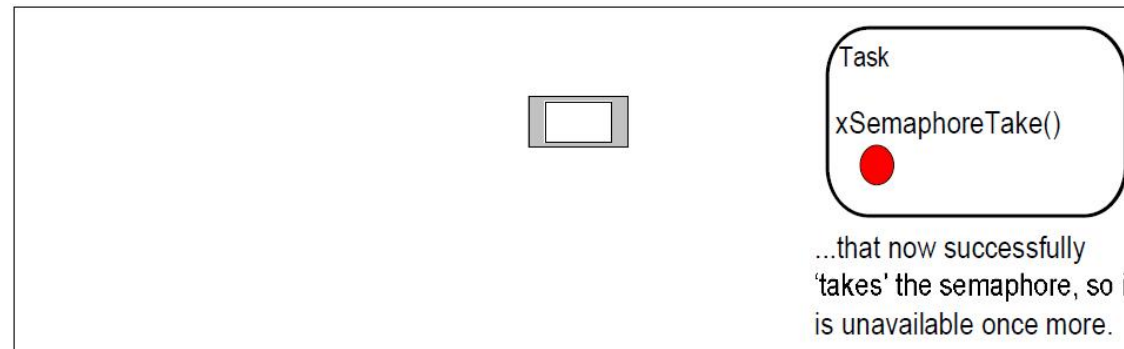


Mikäli mutexia odottavan taskin prioriteetti on korkeampi kuin mutexin haltijan prioriteetti, niin haltijan prioriteetti nostetaan odottajan kanssa samalle tasolle.

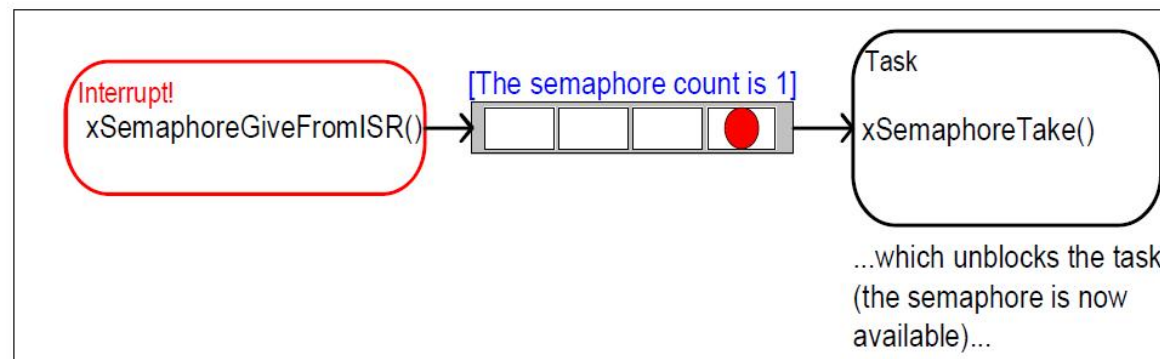
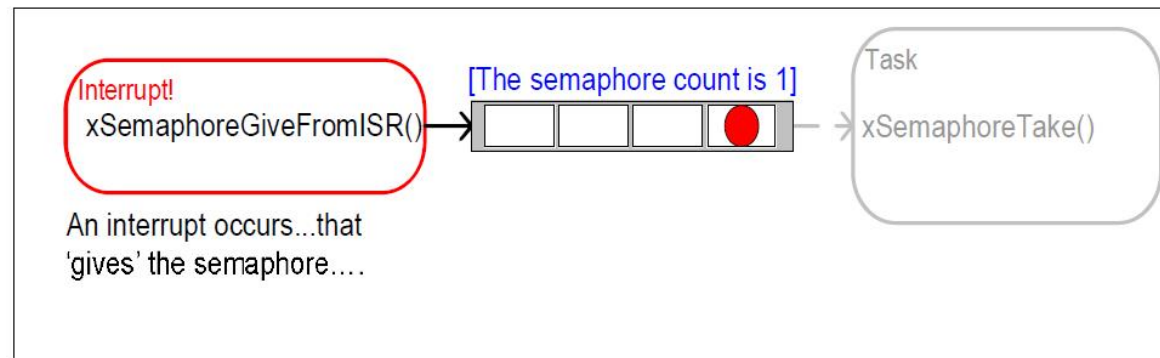
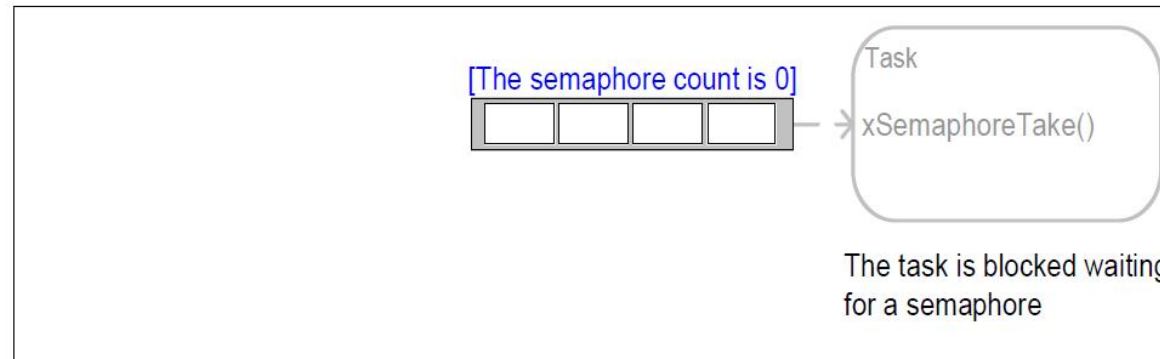
## Taskin ja keskeytyksen synkronointi



## Taskin ja keskeytyksen synkronointi



## Laskevat semaforit



## Laskevat semaforit

