**Kuvaus:**

Kuvassa “errno” globaali muuttuja on asetettu säikeessä 1, jonka jälkeen se yli kirjoitetaan säikeessä 2 ennen kuin säikeen 1 katsonta on suoritettu, jolloin “errno” - muuttujaa lukiessa säikeestä 1 se antaakin säie 2 “errno” sisällön.

Vaikka esimerkkikuvassa sanotaan, että “errno” on globaali muuttuja, se on ohjelmoitu säieystävällisesti

**Ratkaisu:**

* Olla käyttämättä globaaleja muuttujia ja käyttää vain privaatteja globaaleja muuttujia. Tässä tilanteessa, jossa on vain 2 säiettä kilpailemassa globaalista muuttujasta, käyttäisin vain privaattia globaalia muuttujaa.
  + Globaali muuttuja on muuttuja, jota voi käyttää missä tahansa ohjelman vaiheessa
  + Privaatti globaali muuttuja on muuttuja, joka on olemassa vain kyseisen ohjelman osan aikana.

**Vaihtoehtoisia ratkaisuja:**

* Käyttää ns. toisiaan poissulkevaa oliota (mutual exclusion, mutex), joka estää säikeiden käyttävän kyseessä olevaa globaalia muuttujaa samaan aikaan, jolloin säie odottaa omaa vuoroaan toimia muuttujan kanssa.
  + “Kriittinen alue (Critical region)”, joka jättää säikeen odottamaan vuoroaan aktiivisesti. Kömpelö tapa, koska odottelu eskaloituu helposti ja ohjelmasta tulee hidas.
    - “Petersonin ratkaisu”, jossa odotellaan toisten säikeiden valmistumista
    - TSL (Test and Set Lock) ohjeistus: lukitaan 0 ja 1 avulla muuttuja niin, että muut säikeet tietävät saako sitä käyttää vielä. Sopii yhteen monien mutexien kanssa
    - XCHG (eXCHanGe ) ohjeistus: hyvin saman tapainen, kuin TSL, mutta perustuu tiedon vaihtamiseen muistipaikkojen välillä lukituksen tilan mukaan.
  + “Nukahda ja herää”, jossa toinen säie antaa toiselle säikeelle herätyksen, jolloin odottelulta vältytään ja resurssi pysyy vapaana. Ongelmana on, että herättäjäsäie voi antaa nukkuvalle säikeelle käskyn herätä silloin jo, kun nukkuva säie ei ole nukkumistilassa, jolloin herätys hukkuu, eikä kyseinen säie enää unille mentyään herää uudestaan.
    - “Semafori”, ei negatiivinen integeri, jota voi käyttää “nukahda ja herää”-tavassa.