

Tietokoneen toiminnan jatkokurssi, kesä 2020

Harri Kähkönen

Viikkotehtävä 2

Palauta vastauksesi tähän viikkotehtävään kurssin Moodle-oppimisympäristössä tiistaina 11.8.2020 kello 13.59 mennessä pdf-tiedostona. Huomioithan, että sinun on oltava ilmoittautunut viimeistään edellisenä päivänä, jotta pääset Moodleen. Tehtävien ratkaisut käydään läpi ohjatusti Zoomissa tiistaina 11.8.2020 kello 14.15 – 16.00 ja 16.15 – 18.00. Voit osallistua niistä kumpaankin halutessasi. Zoom-linkki on saatavana tietoturvasyistä vain Moodlessa. Muistattehan vastatessa, että yliopistossa niin kokeissa kuin harjoitustehtävissä on plagiointi kielletty, eli esimerkiksi kurssimateriaalista tai muualta suoraan kopioitu tai vähäisesti muunneltu vastaus ei ole sallittu. Opiskeluun yhdessä kannustetaan, mutta kunkin tulee palauttaa itsenäinen tuotos, ei kopiota toisen vastauksista.

Tässä viikkotehtävässä tutustutaan muun lisäksi myös alustavasti ttk-91 -ohjelmointiin. Ttk-91 -ohjelmoinnin oppimistavoitteet on esitelty mooc-materiaalissa kohdassa <https://tietokoneen-toiminnan-jatkokurssi.mooc.fi/luku-5/5-titokone-titotrainer>. Tavoitteiden saavuttamista varten on hyödyllistä tehdä Titotrainer-tehtävät vaikeusasteeltaan 100 - 660 tasoilla a – e. Voit tehdä niitä omassa tahdissasi. Ohjelmointiosaaminen tentitään kokeessa. Olen kirjoittanut ttk-91-oppaan tänne: https://github.com/epicharri/tito/blob/master/learning_materials/ttk-91ohjeita.md#opas-ttk-91-ohjelmointiin.

1. Suoritin ja muisti

- (a) TTK-91-esimerkkisuorittimen rakenteesta on kuva täällä <https://tietokoneen-toiminnan-perusteet.mooc.fi/luku-2/1-suorittimen-rakenne> ja täällä <https://tietokoneen-toiminnan-jatkokurssi.mooc.fi/luku-5/1-ttk-91>. Selosta materiaalin avulla mitä kaikkia osia kuvan mukaiseen ttk-91-suorittimeen kuuluu sekä selosta lyhyesti kunkin tarkoitus. Selitä siis mitä eri rekisterit, yksiköt, väylät jne tekevät tai mitä varten ne ovat. Sisällytä vastaukseen myös mikä on muisti ja mitkä rekisterit liittyvät siihen.

Selosta asiat lyhyesti ja ytimekkäästi omin sanoin. Älä kopioi selityksiä suoraan materiaalista (se on plagiointina kiellettyä eikä edes edistä oppimistasi). Voit ajatella sitä niinkin, että jos kokeeseen sattuisi tulemaan kysymys, jossa hyödynnät tätä tekemääsi kertausmateriaalia, et halua ottaa riskiä tenttisuorituksen hylkäämisestä plagioinnin takia. Älä kirjoita esseemuotoisesti, vaan ennemmin niin kuin kirjoittaisit kertausmateriaalia tai ”lunttilappua” itsellesi tai yrittäisit opettaa asian toiselle. Kirjaa muutama keskeinen asia kustakin. Voit käyttää väliotsikoita, otsikon paksuntamista, ranskalaisia viivoja ym. tehokeinoja. Jos sinulla on opiskelukaveri, tehkää yhdessä ja selittäkää asiat toisillenne! Muistakaa kuitenkin tehdä itsenäinen tuotos, joka ei ole samanlainen kummallakin.

2. Käskyjen nouto- ja suoritusyikli

- (a) Mitkä vaiheet kuuluvat käskyjen nouto- ja suoritusyikliin, kun käytössä on myös keskeytykset?
- (b) Miksi on parempi, että PC:n arvoa kasvatetaan heti käskyn noudon jälkeen eikä vasta käskyn suorittamisen jälkeen?

- (c) Mikä on konekäskyn MUL R1, =2 kenttien (operaatiokoodi, vasen rekisteri, osoitusmoodi, oikea rekisteri, vakio/osoitekenttä) arvot? Mikä on tämän konekäskyn numeerinen esitysmuoto 32-bittisenä binäärilukuna? (Katso <https://tietokoneen-toiminnan-jatkokurssi.mooc.fi/luku-5/1-ttk-91> ja operaatiokoodeista tarkemmin täältä: https://www.cs.helsinki.fi/group/titokone/ttk91_ref.fi.html). Aja Titokoneella tai Titotrainerissa ohjelma, jossa on tuo käsky ja lopussa SVC SP, =HALT, niin voit tarkistaa menikö oikein.
- (d) Oletetaan, että konekäsky MUL R1, =2 sijaitsee ohjelman sisäisen muistiavaruuden osoitteessa 20. Base-rekisterin arvo on 1000 ja Limit-rekisterin arvo on 500. Rekisterin R1 arvo on ennen tuon käskyn suoritusta 10. Ohjelman laskurin PC arvo on nyt 20, edellinen konekäsky on suoritettu ja mahdolliset keskeytykset on käsitelty. Kerro täsmälleen, mitä seuraavaksi tapahtuu suorittimen eri yksiköissä, rekistereissä, väylissä, johtimissa ym. siihen hetkeen asti, kun konekäsky MUL R1, =2 (tai tarkemmin ottaen edellisessä kohdassa laskemasi mukainen kokonaislukuesitys kyseisestä konekäskystä) on noudettu käskyrekisteriin IR. (Katso esimerkkejä sivulla <https://tietokoneen-toiminnan-perusteet.mooc.fi/luku-2/2-suoritussykli>.)
- (e) Jatkoa edelliseen. Oletetaan, että konekäsky MUL R1, =2 on juuri noudettu käskyrekisteriin IR (Instruction Register). Kerro täsmälleen mitä tapahtuu suorittimen eri yksiköissä, rekistereissä, väylissä, johtimissa ym. siihen asti, että kertolaskun tulos on rekisterissä R1. (Katso esimerkkejä sivulla <https://tietokoneen-toiminnan-perusteet.mooc.fi/luku-2/2-suoritussykli>.)
- (f) Oletetaan, että symbolin taulukko arvo on 30. Mikä on konekäskyn ADD R1, taulukko(R5) kenttien (operaatiokoodi, vasen rekisteri, osoitusmoodi, oikea rekisteri, vakio/osoitekenttä) arvot? Mikä on tämän konekäskyn numeerinen esitysmuoto 32-bittisenä kokonaislukuna? (Katso <https://tietokoneen-toiminnan-jatkokurssi.mooc.fi/luku-5/1-ttk-91> ja operaatiokoodeista tarkemmin täältä: https://www.cs.helsinki.fi/group/titokone/ttk91_ref.fi.html.)
- (g) Oletetaan, että yllä mainittu konekäsky ADD R1, taulukko(R5) sijaitsee tuon yllä mainitun ohjelman sisäisen muistiavaruuden osoitteessa 21. Ohjelman laskurin PC arvo on nyt 21, edellinen konekäsky on suoritettu ja mahdolliset keskeytykset on käsitelty. Kerro täsmälleen, mitä tapahtuu suorittimen eri yksiköissä, rekistereissä, väylissä, johtimissa ym. siihen asti, että konekäsky on noudettu käskyrekisteriin IR. (Katso esimerkkejä sivulla <https://tietokoneen-toiminnan-perusteet.mooc.fi/luku-2/2-suoritussykli>.)
- (h) Oletetaan, että yllä mainittu konekäsky ADD R1, taulukko(R5) on juuri noudettu käskyrekisteriin IR. Kerro täsmälleen, mitä tapahtuu suorittimen eri yksiköissä, rekistereissä, väylissä, johtimissa ym. siihen asti, että toinen operandi on saatu noudettua väliaikaisrekisteriin TR (Temporary Register). (Algoritmi on kuvattu kohdassa ”Esimerkki: Ttk-91 käskyn toisen operandin arvon nouto TR:ään, toteutus” sivulta <https://tietokoneen-toiminnan-perusteet.mooc.fi/luku-2/3-konekaskyt>.)
- (i) Jatkoa edelliseen. Kerro täsmälleen, mitä seuraavaksi tapahtuu suorittimen eri yksiköissä, rekistereissä, väylissä, johtimissa ym. siihen asti, että yhteenlaskun tulos on rekisterissä R1. (Katso esimerkkejä sivulla <https://tietokoneen-toiminnan-perusteet.mooc.fi/luku-2/2-suoritussykli>.)

3. Koodinlukua, symbolit, symbolitaulu ja muuttujat

Tarkastellaan seuraavaa ttk-91 -ohjelmaa

```
tbl DS 20
tblKoko EQU 20
x DC 2
Seuraava LOAD R2, =0
          LOAD R1, R2
          MOD R1, x
          STORE R1, tbl(R2)
          ADD R2, =1
          COMP R2, =tblKoko
          JLES Seuraava
          SVC SP, =HALT
```

- (a) Mitä ohjelma tekee taulukolle tbl? Mitkä ovat taulukon alkioiden arvot ohjelman suorituksen jälkeen? Minkä perusteella arvot on laskettu?
- (b) Selitä rivi riviltä lyhyesti, mitä ohjelmassa tapahtuu. Esimerkiksi: LOAD R2, =0 ; Ladataan rekisteriin R2 arvo 0.
- (c) Montako konekäskyä ohjelman suorituksen aikana suoritetaan? Perustele.
- (d) Mitä eroa on symbolin arvolla ja muuttujan arvolla?
- (e) Käsitteet symboli ja symbolitaulu sekä maistiaisia kääntämisestä: Tutustu Tietokoneen toiminnan jatkokurssin kohtiin Symbolitaulu ja Assembler kääntäminen täällä <https://tietokoneen-toiminnan-jatkokurssi.mooc.fi/luku-9/2-kaantaminen>. Kirjoita kohdan (b) ohjelmasta symbolitaulu, jossa kullakin rivillä on symboli ja symbolin arvo. Tee rivi jokaista symbolia kohden (muuttujat, vakiot, rivitunnukset jne), ei kuitenkaan käskyjen ja rekistereiden symboleita. Lisää konekäskyn rakenteesta voi lukea täältä: https://www.cs.helsinki.fi/group/titokone/ttk91_ref_fi.html. Tarkista lopuksi symbolitaulun oikeellisuus ajamalla ohjelma Titokoneessa. (Symbolin HALT arvo on annettu valmiiksi, se on 11.) *Huom! Opettele tekemään tämä kynällä ja paperilla (tai tekstinkäsittelyohjelmalla) sen sijaan että vain katsoisit lopputuloksen Titokoneessa, jotta opit paremmin. Sinun ei tarvitse tehdä vastauksesssi allaolevan kuvan mukaista reunoja varustettua taulukkoa, vaan voit tehdä sen myös ilman noita reunoja.*

Symboli	Arvo
HALT	11
...	

Symboli	Arvo
HALT	11

- (f) Paljonko tämä ohjelma tarvitsee tilaa eli kuinka monta muistipaikkaa ohjelmalle on vähintään varattava? Missä rekisterissä säilytetään ohjelman suoritusaikana tietoa tästä ohjelman koosta? Ota tässä huomioon, että konekäsky `svc sp, =halt` lisää pinoon rekisterin PC arvon ja rekisterin FP arvon, joten tilaa tarvitaan 2 muistipaikkaa enemmän kuin koodi ja data vievät. (Huomio: Jos pinoa käytettäisiin muuten ohjelmassa, pitäisi varautua suurempaan määrään tilaa.)
- (g) Anna vastauksessa tämän ohjelman koodi nyt siten, että konekäskyissä kunkin symbolin tilalla on symbolin arvo. Esimerkiksi jos koodissa olisi konekäsky `ADD R5, B` ja symbolin B arvo olisi 50, kirjoittaisit tuon konekäskyn `ADD R5, 50`. Säilytä kuitenkin noista kääntäjälle annetuista ohjauskäskyistä taulukon ja muuttujan varauskäskyt. Testaa vielä halutessasi Titokoneella, että ohjelma toimii kun tuo muutos on tehty.
- (h) Oletetaan, että ohjelmasta luotu prosessi on sijoitettu keskusmuistiin alkaen osoitteesta 10000. Missä keskusmuistin osoitteessa muuttujan x arvo sijaitsee? Entä missä keskusmuistin osoitteissa taulukon `tbl` arvot sijaitsevat? Mikä pitää rekisterin LIMIT arvon olla vähintään tuota ohjelmaa varten? Mikä on rekisterin BASE arvo?
- (i) Oletetaan, että `ttk-91` -koneeseen olisi toteutettu keskeytyskäsitteijä. Keskeytyskäsitteilyssä aiheutuu tarve laittaa pinoon arvoja. Oletetaan, että pinoa varten varataan runsaasti lisää tilaa, esimerkiksi 40 muistipaikkaa lisää. Kesken ohjelman suorituksen tulee keskeytys, käyttöjärjestelmä saa suoritusvuoron ja päättää siirtää tuon prosessin alkamaan keskusmuistin osoitteesta 9000. Mitä käyttöjärjestelmä tekee saadakseen tämän muutoksen aikaan? Entä tarvitseeko käyttöjärjestelmän tehdä myös ohjelmakoodiin muutoksia? Mitkä ovat muistissa sijainnin siirron jälkeen ohjelmasta luodun prosessin BASE ja LIMIT -rekisterien arvot. Perustelee.

4. Prosessi

Käytä näitä lähteitä tähän vastatessasi: <https://tietokoneen-toiminnan-perusteet.mooc.fi/luku-4/2-prosessi> ja <https://www.cs.helsinki.fi/group/nodes/kurssit/tito/verkkoluennot/pdf/tito8.pdf>.

- (a) Miten prosessi eroaa ohjelmasta?
- (b) Prosessissa on mukana luonnollisesti ohjelman koodi. Mitä muuta prosessille varatussa alueessa on?
- (c) Todellisissa järjestelmissä prosessissa on mukana myös keko, mutta `ttk-91`:ssä ei. Mikä on keko? Mitä varten se on?
- (d) Mitä kaikkea tietoa prosessin kuvaajassa (PCB) säilytetään? Luettele tärkeimmät.

5. Prosessin suoritinympäristö

Yksi prosessin kuvaajan tietueista on prosessin suoritinympäristö. Prosessien vuoronnuksen kannalta se on keskeisessä roolissa. Tutustutaan tähän aiheeseen

tehtävillä, joissa oletetaan, että Tietokoneen toiminta ja Käyttöjärjestelmät -kurssit erinomaisesti suorittanut opiskelija innostui prosessien vuoronnusalgoritmeista ja keskeytyskäsitteilyn hienouksista niin paljon, että ohjelmoi ttk-91 -koneelle upouuden käyttöjärjestelmän nimeltään **Heinix**.

Erityisesti hän halusi siihen tietysti keskeytyskäsitteilyn ja prosessien vuoronnuksen. Prosessin kuvaajia varten Heinixissä on prosessitaulu, joka on toteutettu taulukkona eli tietorakenteena, jossa perättäisissä muistipaikoissa (ttk-91:ssä on käytössä sanaosoitteet ja sana on tässä yhteydessä 4 tavua) on kunkin prosessin kuvaajan (PCB) osoite keskusmuistissa ja prosessin id (PID) toimii indeksinä. Prosessitaulu on siis toteutettu tavanomaisena ttk-91:n taulukkona. Prosessitaulu sijaitsee keskusmuistin osoitteessa 8192 (0x2000). Taulun koko on 256 muistipaikkaa. Prosessitaulun Heinixin kehittäjä alusti siis käyttöjärjestelmän lähdekoodissa näin:

```
procTbl DS 256
```

Prosessin kuvaaja on eräs tärkeä käyttöjärjestelmän tietorakenne. Se on tietue, joka sisältää eri kenttiä (yksi muistipaikka) ja tietueita (enemmän kuin yksi muistipaikka). Siis yksinkertaisesti taulukko, jossa on perättäisissä muistipaikoissa talletettuna tarvittavat tiedot. Kunkin prosessin kuvaaja on talletettu käyttöjärjestelmän päättämään paikkaan keskusmuistissa. Heinixissä jokaista prosessi id:tä kohden on etukäteen varattu muistialue. Kun käyttöjärjestelmä luo uuden prosessin, se tarkistaa ensin prosessitaulusta, onko vapaita prosessin kuvaajia ja jos on, se alustaa prosessin kuvaajan prosessitaulun ilmaisemalle muistialueelle. Vain etuoikeutetussa tilassa on mahdollista päästä lukemaan ja kirjoittamaan prosessitauluun tai prosessikuvaajien muistialueelle.

Heinixissä kunkin kuvaajan ensimmäisessä kentässä (indeksi 0 eli muistipaikka 0 kuvaajan alkuosoitteesta laskettuna) on tallennettuna PID. Prosessin suoritinympäristön tietue sijaitsee muistipaikoissa 3 – 14 kuvaajan alkuosoitteesta laskettuna.

- (a) Oletetaan, että Heinixin hallinnoimalla (simuloidulla) ttk-91-suorittimella on ajossa prosessi P, jonka id on 172. Kelloaitekeskeytyksen johdosta tullaan tilanteeseen, jossa suorituksessa oleva prosessi vaihdetaan ja käyttöjärjestelmä (sen dispatcher-rutiini) tallettaa prosessin P suoritinympäristön prosessin kuvaajaan. Mitkä kaikki tiedot ttk-91 -suorittimessa kuuluvat prosessin suoritinympäristöön ja siten pitää tallettaa prosessin kuvaajaan? Luettele siis täsmälleen nuo kaikki tiedot. Miksi juuri nuo tiedot pitää tallettaa prosessin kuvaajaan? Luettele myös ne rekisterit, joita ei tarvitse tallettaa prosessin kuvaajan suoritinympäristötietueeseen ja selitä miksi niitä ei tarvitse tallettaa. (Ylimääräinen lisähuomio: kun käyttöjärjestelmä on tehnyt tuon tallennuksen, se siirtää tämän prosessin osoittimen (*pointer*) valmis suoritukseen -jonoon. Heinixissä tämä jono on toteutettu tietorakenteena jota kutsutaan linkitetyksi listaksi.)
- (b) Ttk-91-simulaattorissa ja sen käskykannassa ei ole kaikkia konekäskyjä, joita keskeytyskäsitteilyn mahdollistaminen edellytti, joten Heinixin kehittäjä ohjelmoi uuden ttk-91-simulaattorin uusine käskyineen ja niiden simuloineineen. Oletetaan, että kehittäjä ei muuttanut ttk-91-konekäskyn rakennetta, vaan sen sijaan lisäsi uusia käskyjä ttk-91-suorittimen käskykantaan. Miksi uusien konekäskyjen lisääminen on mahdollista ilman konekäskyn rakenteen muutosta? (Vinkki: *montako konekäskyä nyt on ja paljonko niitä mahtuu. Käskykanta löytyy täältä:* https://www.cs.helsinki.fi/group/titokone/ttk91_ref.fi.html.)

- (c) Mitkä suoritinympäristön tiedot pystytään tallettamaan ja vastaavasti palauttamaan suorittimelle ttk-91-suorittimen nykyisellä käskykannalla? Entä mitä lisäyksiä Heinixin kehittäjän oli pakko tehdä ttk-91-suorittimen käskykantaan prosessin suoritinympäristön tallennuksen ja palautuksen mahdollistamiseksi sekä keskeytyskäsittelyn mahdollistamiseksi? (Tätä tehtävää varten käy läpi Ttk-91 käskykanta ja tutki mitkä noista suoritinympäristön tiedoista pystytään nykyisellään toteuttamaan ja mitkä ei. Käskykanta löytyy täältä: https://www.cs.helsinki.fi/group/titokone/ttk91_ref.fi.html.)
- (d) Jatkoa edelliseen. Saako noita uusia konekäskyjä käyttää muuten kuin etuoikeutetussa tilassa? Perustele.
- (e) Jatkoa edelliseen. Laitettuaan prosessin P valmis suoritukseen -jonoon, käyttöjärjestelmä valitsee samasta jonosta seuraavaksi suoritukseen prosessin Q, jonka id on 227. Miten käyttöjärjestelmä (sen vuorontajan dispatcher-rutiini) löytää prosessin Q suoritinympäristön tiedot päästäkseen siirtämään tiedot suorittimen rekistereihin? Oleta tässä, että prosessin Q id on jo ladattu rekisteriin R1 ja prosessitaulun alkuosoite on procTbl. Kuvaile konekäskyinä tai selittäen vaiheet.
- (f) Jatkoa edelliseen. Mitkä prosessin Q suoritinympäristön tiedot ladataan suorittimelle ensiksi ja mikä viimeisenä? Mistä käskystä prosessin Q suoritus alkaa? Missä vaiheessa suoritusvuoro siirtyy prosessille Q?
- (g) Jatkoa edelliseen. Kerro mitkä olivat suorittimen tilat (käyttäjätila, etuoikeutettu tila) ja missä vaiheessa ne muuttuivat alkaen tilanteesta, jossa P oli suorituksessa ja tapahtui kellolaitekeskeytys, siihen hetkeen, kun prosessin Q ensimmäinen käsky suoritetaan?

6. Keskeytykset

- (a) Miksi järjestelmässä kannattaa olla keskeytyksiä? Kerro kolme eri keskeytystyyppiä ja anna kustakin esimerkki.
- (b) Mitä suorittimessa tapahtuu, jos käskyn suorituksen yhteydessä tulee käskyn aiheuttama virhetilanne, esimerkiksi nollalla jako tai kokonaisluvun ylivuoto?
- (c) Miten suoritin tietää käskyn suorituksen jälkeen, että on tarve keskeytykselle? Entä kun se huomaa sen, mitä suoritin tekee? Vastaa vaihe vaiheelta omin sanoin siihen saakka, kun kontrolli siirtyy asianmukaiselle keskeytyskäsittelijälle. (Katso <https://tietokoneen-toiminnan-perusteet.mooc.fi/luku-2/2-suoritusytkli> kohta ”Käskyjen nouto- ja suoritusytklin toteutus laitteistossa”.)
- (d) Kun asianmukainen keskeytyskäsittelijä aloittaa toimintansa, mitä se tekee ensimmäiseksi? Entä miten keskeytyskäsittelijä hoitaa palaamisen takaisin keskeytettyyn ohjelmaan?
- (e) Missä vaiheessa on tärkeää estää keskeytysten käsittely? Miten tämä käytännössä toteutetaan suorittimessa? Miksi on haitallista, jos keskeytykset ovat estettynä pitkään?
- (f) Jos käskyä suorittaessa on yritetty nollalla jakamista ja suorittimen tehtyä kohdassa b ja c selittämäsi asiat, niin mitä keskeytyskäsittelijä tekee?

- (g) Mitkä tiedot suoritinympäristöstä tallennetaan laitteistotasolla? Miksi nämä täytyy tallentaa laitteistotasolla (todellisessa järjestelmässä)? Entä mitkä tallennetaan ohjelmistotasolla keskeytyskäsitteijän toimesta?
(Muistutuksena, että ttk-91-suoritin on simuloitu suoritin, eli kaikki siinä tehtävät toimenpiteet tehdään korkean tason kielellä simuloiden todellista laitteistotason toimintaa. Siinä on siis kuitenkin kuvitteellisesti laitteistotaso eli simulaattoriohjelman tekemät toiminnot ja ohjelmistotasona ttk-91 -konekielen konekäskyt.)
- (h) Oletetaan edellisen tehtävän *Heinix*-käyttöjärjestelmän olemassaolo. Mitä uusia etuoikeutettuja konekäskyjä Heinixin kehittäjän täytyi lisätä ttk-91-simulaattorin käskykantaan, jotta keskeytyskäsitteily tämän tehtävän mukaisella tavalla on mahdollinen toteuttaa? Mihin toimintoihin ei tarvittu uusia konekäskyjä?

7. Prosessin tilat

Täällä <https://tietokoneen-toiminnan-perusteet.mooc.fi/luku-4/2-prosessi> ja täällä <https://www.cs.helsinki.fi/group/nodes/kurssit/tito/verkkoluennot/pdf/tito8.pdf> on esitetty prosessin 5-tilainen elinkaarimalli. Anna seuraaville tilanteille yksi tai kaksi erilaista esimerkkiä. Kerro mikä aiheutti tarpeen tilasiirtymälle ja mitä tilanmuutoksessa tapahtuu käyttöjärjestelmän tietorakenteissa. Mitä tietoja kopioidaan, mistä ja minne? Entä jonot?

- (a) Prosessi R siirtyy luonti-tilasta odottaa-tilaan.
- (b) Prosessi R siirtyy suorituksessa-tilasta odottaa-tilaan.
- (c) Prosessi R siirtyy valmis suoritukseen -tilasta suorituksessa-tilaan.
- (d) Prosessi R siirtyy suorituksessa-tilasta valmis suoritukseen -tilaan.
- (e) Prosessi R siirtyy poistettu tai tapettu -tilaan.