

## Oppimistavoitteet kurssilla Rinnakkaisohjelmointi

Pääteema	Esitiedot	Lähestyy oppimistavoitteita	Saavuttaa oppimistavoitteet	Syventää oppimistavoitteita
Rinnakkaisuus ja rinnakkaisuuden soveltaminen tietojenkäsittelyjärjestelmissä	<p>Kurssin <i>Tietokoneen toiminta</i> perusteella ymmärtää, miten ohjelman suoritus tietokoneessa etenee:</p> <ul style="list-style-type: none"> <li>• miten prosessori suorittaa ohjelman käskyjä</li> <li>• miten käyttöjärjestelmä käyttää prosesseja ohjelmien suorituksen ohjaukseen</li> <li>• miten prosessien rinnakkainen eteneminen perustuu laitteistokeskeytyksiin ja käyttöjärjestelmän skedulointipäätöksiin.</li> </ul> <p>Oleellista: osaa selittää, miten kaksi ohjelmaa voi edetä konekäskytasolla rinnakkain.</p>	<p>Tietää, mitkä tekijät johtavat rinnakkaisiin toimintoihin laitteisto- ja ohjelmistotasolla.</p> <p>Tuntee esimerkkejä rinnakkaisuuden esiintymistä laitteisto-, käyttöjärjestelmä-, sovellusohjelma- ja verkkosovellustasoilla.</p> <p>Osaa kuvata esimerkkejä käyttäen rinnakkaiseen suoritukseen liittyvät peruskäsitteet:</p> <ul style="list-style-type: none"> <li>• keskinäinen poissulkeminen ja suorituksen kriittinen vaihe</li> <li>• prosessien synkronointi</li> <li>• lukkiutuminen</li> </ul> <p>Tuntee näiden taustalla olevat tavoitteet ja osaa antaa esimerkkejä ongelmista, joihin näiden virheellinen ratkaisu tai ratkaisematta jääminen voi johtaa.</p>	<p>Ymmärtää prosessien suorituksen etenemisen epädeterministisen luonteen ja sen vaikutukset.</p> <p>Osaa selittää, rinnakkaisuuden peruskäsitteiden taustalla olevat tavoitteet.</p> <p>Ymmärtää toiminnallisen synkronoinnin ja ehtosynkronoinnin välisen yhteyden.</p> <p>Osaa selittää prosessien käyttäytymisen perusteella rinnakkaisuuden liittyvien ongelmien synnyn.</p> <p>Ymmärtää rinnakkaisuusongelmien ratkaisuille asetettavat vaatimukset.</p> <p>Ymmärtää laitteistotason arkkitehtuurin vaikutuksen siihen, mitä voidaan rinnakkaistaa ja mitä hyötyä rinnakkaistamisesta voi olla.</p>	<p>Ymmärtää, miten käyttöjärjestelmätasolla rinnakkaisuutta hallitaan.</p> <p>Ymmärtää rinnakkaisuuden erilaisen luonteen laitteisto-, ohjelmisto- ja verkkotasojen toiminnoissa.</p> <p>Tietää, miten hajautus eroaa rinnakkaisuudesta.</p>

Pääteema	Esitiedot	Lähestyy oppimistavoitteita	Saavuttaa oppimistavoitteet	Syventää oppimistavoitteita
Rinnakkaisohjelmoinnin peruskäsitteistö ja rakennemallit		<p>Osaa selittää, mitä tarkoittavat peruskäsitteet</p> <ul style="list-style-type: none"> <li>• synkroninen / asynkroninen</li> <li>• atominen.</li> </ul> <p>Tuntee käyttäytymismallit</p> <ul style="list-style-type: none"> <li>• tuottaja-kuluttaja</li> <li>• asiakas-palvelin</li> <li>• vertaissolmujen verkko</li> <li>• puomisynkronointi</li> <li>• aktiivipalvelin ja kohtaaminen (rendezvous)</li> <li>• lukijat-kirjoittajat (epäsymmetriset luokkamallit)</li> </ul> <p>Osaa lukea näitä malleja toteuttavia algoritmeja ja osaa selittää, miten ko. ohjelmat eri tilanteissa toimivat .</p> <p>Tietää, mitkä (kurssilla myöhemmin esiteltävät) ohjelmointikielten käsitteet soveltuvat kommunikointiin, jos prosessit ovat samassa / eri koneessa.</p>	<p>Osaa soveltaa kaikkia näitä malleja erilaisissa ongelmanratkaisutilanteissa erilaisia (kurssilla myöhemmin esiteltäviä) ohjelmointikielten käsitteitä käyttäen.</p> <p>Pystyy perustelemaan näitä malleja soveltavissa ratkaisuissa ratkaisun oikeellisuuden.</p> <p>Ymmärtää, miten kussakin mallissa toiminta rinnakkaistuu tai sarjallistuu.</p>	<p>Osaa arvioida erilaisien rakennemallien käyttökelpoisuutta erilaisilla laitteisto- ja ohjelmisto-alustoilla.</p> <p>Osaa arvioida, missä määrin tietty sovellus on rinnakkaistettavissa.</p>
Lukkumuuttajat	Tuntee käyttöjärjestelmän toiminnan prosessien skeduloinnissa.	<p>Osaa toteuttaa poissulkemisen käyttäen lukkoa ja test-and-set –käskyä.</p> <p>Tietää, että lukkomuuttujan käyttö johtaa aktiiviodotukseen ja tietää, mitä haittaa tästä on.</p>	Tietää, missä eri tilanteissa lukkomuuttujan ja aktiiviodotuksen käyttö on perusteltua.	

Pääteema	Esitiedot	Lähestyy oppimistavoitteita	Saavuttaa oppimistavoitteet	Syventää oppimistavoitteita
<p>Prosessien välinen vuorovaikutus: semaforit ja monitorit</p>	<p>Tuntee säikeiden yhteisen muistinkäytön.</p>	<p>Osaa selittää semaforin ja monitorin rakenteen sekä vastaavien operaatioiden aiheuttaman toiminnallisuuden.</p> <p>Osaa käyttää semaforeja ja monitoreja eri perusmallien ratkaisuun (tuottajat-kuluttajat, asiakas-palvelin, puomisynkronointi).</p> <p>Ymmärtää algoritmeja, joissa on käytetty halkaistuja semaforeja, yksityisiä semaforeja, "viestikapulan luovutus" –tekniikkaa ja "ehdon säilytyksen" tekniikkaa.</p> <p>Osaa selittää lukijat-kirjoittajat –ongelman ratkaisun toiminnan.</p>	<p>Osaa laatia semaforeja ja monitoreja käytäviä algoritmeja, joissa hyödynnetään aiemmin mainittuja perusmalleja. Osaa toteuttaa näitä Java-ympäristössä.</p> <p>Osaa toteuttaa puomisynkronoinnin. Osaa säädellä resurssien jakelun järjestystä (satunnainen, FIFO, prioriteettipohjainen).</p> <p>Osaa toteuttaa käyttäjäluokkaperustaisen synkronoinnin (lukijat-kirjoittajat).</p> <p>Osaa toteuttaa aktiivipalvelimen ("nukkuvaa parturi").</p> <p>Osaa perustella ratkaisujen toimivuuden ehtojen säilyvyyttä käyttäen.</p> <p>Ymmärtää keskitettyjen (jakelijaperustisten) ja hajautettujen (sopimisperustisten) ratkaisujen samankaltaisuudet ja eroavuudet.</p> <p>Ymmärtää, miksi lukkoja, semaforeja ja resurssikohtaisia varusalgoritmeja käytetään aivan erilaisiin resurssinvaraustarpeisiin.</p>	<p>Osaa toteuttaa semaforin ja monitorin lukkumuuttujia käyttäen.</p> <p>Osaa perustella monimutkaisten rinnakkaisalgoritmien toimivuuden.</p> <p>Osaa suunnitella ja toteuttaa yhdessä koneessa toimivia rinnakkaissovelluksia.</p>

Pääteema	Esitiedot	Lähesty oppimistavoitteita	Saavuttaa oppimistavoitteet	Syventää oppimistavoitteita
Lukkiutuminen	Tuntee käyttöjärjestelmän toiminnan resurssien jakelijana.	<p>Osaa kuvata esimerkkejä käyttäen lukkiutumistilanteen synnyn.</p> <p>Osaa selittää välttämättömät ja riittävät ehdot lukkiutumiselle.</p> <p>Osaa kuvata tapoja lukkiutumisen havaitsemiseen (?), estämiseen ja välttämiseen.</p> <p>Tuntee "aterioivien filosofien" ongelman.</p>	<p>Ymmärtää pankkiirin algoritmin toimintaperiaatteen ja osaa kuvata algoritmin toimintaa esimerkkejä käyttäen.</p> <p>Tietää lukkiutumisen käsittelymenetelmien rajalliset mahdollisuudet todellisissa ympäristöissä.</p> <p>Osaa toteuttaa aterioivien filosofien algoritmin sekä keskitettynä (resurssin jakelijaa käyttäen) että hajautettuna.</p>	Ymmärtää lukkiutumisen riskin toisaalta käyttöjärjestelmäympäristössä (pieni) ja toisaalta tietokanta- ja hajautetuissa ympäristöissä (suuri).
Prosessien välinen kommunikointi: sanomanvälitys	Tietää, että käyttöjärjestelmä voi välittää sanomia prosessilta toiselle.	<p>Osaa selittää, miten sanomanvälitys kanavien kautta toimii.</p> <p>Tietää synkronisen ja asynkronisen sanomanvälityksen eron.</p> <p>Osaa käyttää sanomanvälitystä perusmallien ratkaisuun (tuottajakuluttaja, asiakas-palvelin).</p> <p>Huomaa monitoripohjaisen palvelinratkaisun ja sanomanvälitykseen perustuvan palvelinratkaisun samankaltaisuuden.</p>	<p>Osaa toteuttaa rinnakkaislaskennan kaikki perusmallit (ml. suodinmalli, yleinen asiakas-palvelin) algoritmitasolla.</p> <p>Osaa toteuttaa vertaiskommunikointiin perustuvia ratkaisuja.</p> <p>Ymmärtää monitoripohjaisen palvelinratkaisun ja sanomanvälitykseen perustuvan palvelinratkaisun samankaltaisuudet ja eroavuudet.</p>	<p>Osaa yleistää sanomanvälityksen verkkotasolle.</p> <p>Tuntee CSP:n ja tietää, mihin sitä voidaan käyttää.</p>
Prosessien välinen kommunikointi: etäproseduurikutsu (RPC) ja vartioitu rinnakkaisodotus (input statement)	Kurssin <i>Tietokoneen toiminta</i> perusteella tietää, miten aliohjelmia suoritetaan.	<p>Tietää etäproseduurikutsun ja vartioitun rinnakkaisodotuksen toimintaperiaatteet.</p> <p>Osaa algoritmitasolla toteuttaa asiakas-palvelimen (RPC) ja aktiivisen palvelimen kohtaamisen (rendezvous).</p> <p>Ymmärtää erillisen rajapintamäärittelyn tarpeen.</p>	<p>Osaa selittää, miten etäproseduurikutsu ja vartioitu rinnakkaisodotus toimivat.</p> <p>Tuntee etäproseduureihin liittyvät virhesemantiikat.</p> <p>Osaa toteuttaa algoritmitasolla rinnakkaisen odotuksen sekä käyttäen säikeitä että käyttäen vartioituja odotuslauseita.</p>	

Pääteema	Esitiedot	Lähestyy oppimistavoitteita	Saavuttaa oppimistavoitteet	Syventää oppimistavoitteita
Rinnakkaisohjelmointi	Ohjelmointikursseilla opitut perustiedot ja -taidot	Osaa laatia pieniä rinnakkaisuuden hallintaa vaativia algoritmeja käyttäen proseduraalista algoritmien kuvauskieltä. Tietää tavallisimmat rinnakkaisohjelmien virhetyypit.	Osaa laatia pieniä rinnakkaisuuden hallintaa vaativia ohjelmia Java-ympäristössä. Osaa päätellä rinnakkaisalgoritmin oikeellisuuden.	Tuntee Javan ja C:n rinnakkaisuuden hallinnan menetelmien periaatteet ja osaa toteuttaa näillä kielillä rinnakkaisohjelmia. Osaa testata rinnakkaisohjelmia. Osaa todistaa rinnakkaisalgoritmin oikeellisuuden

Termi ”ymmärtää” tarkoittaa sitä, että henkilö pystyy perustellen selittämään asiaa tuntemattomalle, mistä on kysymys, mikä on oleellista ja millaisia erilaisia syy-seuraussuhteita asioiden välillä vallitsee.