



FI000091026B

**(B) (11) KUULUTUSJULKAISU
UTLAGGNINGSSKRIFT****91026****C (10) Patentti myönnetty**
Patent beviljat 05.01.1991

(51) Kv.1k.5 - Int.cl.5

G 06F 9/46**SUOMI-FINLAND****(FI)****Patentti- ja rekisterihallitus**
Patent- och registerstyrelsen

(21) Patenttihakemus - Patentansökning	910479
(22) Hakemispäivä - Ansökningsdag	31.01.91
(24) Alkupäivä - Löpdag	31.01.91
(41) Tullut julkiseksi - Blivit offentlig	01.08.92
(44) Nähtäväksipanon ja kuul.julkaisun pvm. - Ansökan utlagd och utl.skriften publicerad	14.01.94

(71) Hakija - Sökande

1. Valtion teknillinen tutkimuskeskus, Vuorimiehentie 5, 02150 Espoo, (FI)

(72) Keksijä - Uppfinnare

1. Auer, Antti, Ravanderintie 3 A 6, 90570 Oulu, (FI)
2. Levanto, Mikko, Hiidentie 8 A 3, 90550 Oulu, (FI)
3. Okkonen, Jyrki, Puutarhakatu 23 A, 90100 Oulu, (FI)
4. Okkonen, Ari, Tarmontie 45 B 8, 90530 Oulu, (FI)

(74) Asiamies - Ombud: **Papula Rein Lahtela Oy**

(54) Keksinnön nimitys - Uppfinningens benämning

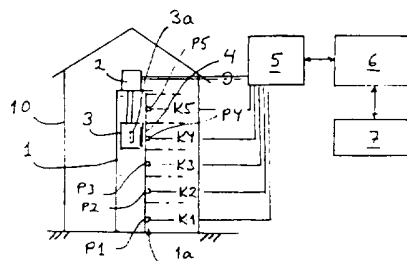
Parannettu menetelmä taskien käsittelymiseksi
Förbättrat förfarande för behandling av uppdrag

(56) Viitejulkaisut - Anförda publikationer

EP A 360897 (G 06F 9/46), EP A 360900 (G 06F 9/46), US A 4482962 (G 06F 9/46),
US A 4636944 (G 06F 9/46), US A 4794526 (G 06F 9/46)

(57) Tiivistelmä - Sammandrag

Keksinnön kohteena on parannettu menetelmä taskien käsittelymiseksi reaaliajassa erityisesti sulautetuissa järjestelmissä. Taskit on varustettu prioriteeteilla ja taskeista kukin suoritetaan aina loppuun siten, että vain korkeampiprioriteettisen taskin suoritus voi sen suoritusta hidastaa. Keksinnön mukaisesti aina kun havaitaan tapahtuma, sen käsittely aloitetaan heti, vaikka edellisen tapahtuman käsittely olisi kesken, ellei sitä ole erityisesti kielletty. Tämä käsittely toteutetaan tietyn käsittelysekvenssin mukaisesti. Se tapahtuma aiheuttaa keskeytyksen, josta syntyy aina tapahtumaa kuvaava viesti ja uusi käsittelysekvenssi heti, kun käynnissä olevassa käsittelysekvenssissä on keskeyttäminen sallittu. Yhdellä käsittelijällä voi tällöin olla useita käsittelysekvenssejä kesken viimeksi aloitetun käsittelysekvenssin ollessa käynnissä. Taskit voidaan varustaa prioriteetin sijasta palveluajoilla ja toteuttaa menetelmä tältä pohjalta vastaavasti.



Uppfinningen avser ett förbättrat förfarande för behandling av tasker i realtid speciellt i fusionerade system. Taskerna är försedda med prioritet och av taskerna utförs alltid var och en till slut så, att endast en högre prioriterad tasks behandling kan göra dess behandling långsammare. Enligt uppfinningen påbörjas alltid behandlingen av en händelses genast, då den observeras, fastän föregående händelses behandling ännu skulle vara oavslutad, ifall detta ej speciellt är förbjudet. Denna behandling förverkligas enligt en bestämd behandlingssekvens. Den händelse åstadkommer ett avbrott, varav alltid uppstår ett händelsen beskrivande meddelande och en ny behandlingssekvens genast, då i den igång varande behandlingssekvensen ett avbrytande är tillåtet. En behandlare kan härvid ha flera oavslutade behandlingssekvenser då den senast påbörjade behandlingssekvensen är igång. Taskerna kan i stället för med prioritet förses med behandlingstider och förfarandet förverkligas på motsvarande sätt utgående från denna grund.

PARANNETTU MENETELMÄ TASKIEN KÄSITTELEMISEKSI

Keksinnön kohteena on itsenäisen patenttivaatimuksen johdanto-osassa määritelty parannettu menetelmä
5 taskien käsitlemiseksi.

Keksinnön mukainen menetelmä soveltuu erityisesti sulautettuihin järjestelmiin. Sulautetuissa järjestelmissä joitakin järjestelmän toiminnallisia ominaisuuksia toteutetaan tietokoneella. Sulautettuja järjestelmiä ovat mm. pankkiautomaatit, radiopuhelimet, hissit
10 jne. Nämä järjestelmät asettavat laitteille ja ohjelmille erilaisia vaatimuksia tavanomaisiin tietokonepohjaisiin ohjausjärjestelmiin nähden.

Sulautetuissa järjestelmissä tietokoneen pitää
15 reagoida riittävällä nopeudella ulkoisiin tapahtumiin, vaikka sillä olisi aikaisemman tapahtuman käsittely kesken. Tehtävien kiireellisyys riippuu tehtävien luonteesta. Paperikoneen ohjaustietokoneen esimerkiksi pitää pystyä antamaan tilanneraportti pyydettyäessä samalla,
20 kun se ohjaa normaalisti käyvää paperikonetta. Useimmiten ulkoiset tapahtumat ovat keskenään tilastollisesti riippumattomat, joten tietokone ei voi suunnitella käsittelyjärjestystä tulevien tapahtumien aikojen perusteella.

25 On lisäksi huomattava, että kullakin tietokoneella on sille ominainen rajoitettu kapasiteetti. Tämä aiheuttaa sen, että kesken olevaa käsittelyä ei aina ole mahdollista saada valmiiksi ennen kiireellisen tapahtuman käsittelyn määräaikaa. Jos olisi mahdollista
30 käyttää rajattoman tehokasta tietokonetta, kaikki tapahtumat voitaisiin käsitellä peräkkäin siinä järjestyksessä kuin ne tapahtuvat.

Edellä esitettyjen ongelmien ratkaisemiseksi on kehitetty useita ratkaisuja. Erään menettelyn (keskeyttämätön skedulointi) mukaan kaikki tapahtumat käsitellään yksi kerrallaan loppuun asti eräajotyyppisesti.
35 Tällä menettelyllä on haittansa. Kiireellisiä tapahtumia

ei voida käsitellä ennen kuin kesken oleva tapahtuman käsittely on suoritettu loppuun. Rajoitusta voidaan kiertää lisäämällä käsittelyjärjestykseen riittävän tiheään kohtia, joissa tarvittaessa käsitellään kiireelliset tapahtumat. Toisaalta on todettava, että tilannetta helpottaa, jos voidaan käyttää tavallista tehokkaampaa prosessoria.

Erään toisen menettelyn mukaisesti keskeytysmenettelyä voidaan käyttää laitetapahtumien käsittelyyn. Tässä järjestelyssä on myös ongelmia. Kun tapahtuman käsittely aloitetaan, se on suoritettava kerralla loppuun, vaikka tuotettavien vasteiden kiireellisyys voi olla hyvin erilainen. Tätä ongelmaa voidaan osittain kiertää suorittamalla käsittelyt loppuun eräajotyyppisesti: kiireelliset vasteet tuotetaan heti keskeytyksen tapahduttua ja samalla asetetaan pyyntö jatkokäsittelystä; kun eräajotyyppisessä käsittelyssä on saatu edellinen työ loppuun, tarkastetaan asetetut pyynnöt ja suoritetaan niiden perusteella jatkokäsittelyt. Tällä tavoin yhdestä tapahtumasta voidaan tuottaa korkeintaan kahden kiireellisyysluokan vasteita.

Eräänä kolmantena menettelynä käytetään monitaskimenettelyä tehtävien rinnakkaiseen suorittamiseen. Taskit ovat päättymättömiä tapahtumankäsittelyrutiineja.

Monitaskimenettelyssä useita tapahtumia voidaan käsitellä yhtäaikaan. Monitaskimenettelyssä voidaan käyttää erilaisia skedulointi- eli töidenjärjestelyperiaatteita. Esimerkkejä skedulointiperiaatteista ovat kiinteä prioriteettiskedulointi, jossa taskeille on määritelty kiinteä kiireellisyysjärjestys, määräaika-skedulointi, joissa taskit suoritetaan tilanteen mukaan kiireellisinsä ensin ja aikajakoskedulointi, jossa eri taskeille annetaan suoritusaikaa vuorotellen, sekä erilaiset näiden yhdistelmät.

Monitaskimenettelyä käytettäessä esiintyy seuraavia ongelmia. Monitaskimenetelmää sovellettaessa kullekin taskille varataan oma muistialue työtilaa

varten. Muistialue täytyy mitoittaa maksimitarpeen mukaan, johon kuuluu myös keskeytyskäsittelyn ja käyttöjärjestelmäpalveluiden vaatima tila. Viimeksimainitut tulevat näin varatuiksi moninkertaisesti. Taskit kutsuvat käyttöjärjestelmää pitääkseen yhteyttä ympäristöönsä. Testausta varten tällaiset tapahtumankäsittelyrutiinit vaativat aidon näköisen käyttöjärjestelmäympäristön ja runsaasti testiapuneuvoja.

Keksinnön tarkoituksena on poistaa edellä mainitut epäkohdat. Keksinnön tarkoituksena on erityisesti tarjota uusi parannettu menetelmä taskien eli tapahtumankäsittelyrutiinien suorittamiseksi reaaliajassa erityisesti sulautetuissa järjestelmissä.

Keksinnön mukaiselle menetelmälle on tunnusomaista se, mitä on esitetty patenttivaatimuksessa 1.

Keksinnön mukaiselle rinnakkaiselle menetelmälle on tunnusomaista se, mitä on esitetty patenttivaatimuksessa 2.

Keksinnön mukaisen parannetun taskien käsittelymenetelmän selostuksen yhteydessä käytetään käsitteitä, jotka määritellään seuraavassa.

Käsittelijä on laite tai sitä vastaava kokonaisuus, joka pystyy toimintaohjeiden mukaan reagoimaan tapahtumiin tuottamalla vasteita. Käsittelijällä on tila, jonka käsittelijä ottaa huomioon tapahtumia käsitellessään ja jota käsittelijä muuttaa käsittelyohjeiden mukaisesti.

Tapahtumia ovat sellaiset muutokset käsittelijän ympäristössä, jotka käsittelijä pystyy havaitsemaan. Tapahtumia ovat esimerkiksi prosessin tilan muuttuminen, joka havaitaan anturien avulla, käyttäjän suorittama ohjaustoimenpide, tai toiselta käsittelijältä saapuva sanoma. Tapahtumaksi lasketaan myös määrääjän kuluminen aikaisemmasta tapahtumasta.

Vasteita ovat käsittelijän ympäristössään tai omassa tilassaan aikaansaamat muutokset. Ympäristöön tuotettuja vasteita ovat esimerkiksi toimilaitteiden

ohjaus, tiedon antaminen käyttäjälle tai sanoman lähettäminen toiselle käsittelijälle. Tilaa muuttava vaste on esimerkiksi siirtyminen seuraavaan vaiheeseen sekvenssiohjauksessa.

- 5 Käsittelijän toiminnallinen rakenne jakaantuu taskeihin eli tapahtumankäsittelyrutiineihin. Käsittelijän tila muodostuu kunkin taskin tilasta ja toimintaohjeet muodostuvat kunkin taskin toimintaohjeista. Tapahtuman käsittely voi jakautua useille taskeille.
- 10 Tapahtumat tuottavat keskeytysmekanismin avulla viestejä taskien käsiteltäväksi. Myös taskit voivat tuottaa viestejä toisilleen. Viesti on tietorakenne, joka vie tietoa jollekin taskille. Taskit käsittelevät viestejä ottaen huomioon oman tilansa. Viestien lisäksi taskit
- 15 voivat tuottaa vasteita käsittelijän ympäristöön ja omaan sisäiseen tilaansa.

- Tapahtuma virittää keskeytyksen, joka laukeaa heti, ellei käsittelijä ole sitä erikseen kieltänyt. Keskeytyksen laukeaminen keskeyttää sen toiminnan, mitä
- 20 käsittelijä on sillä hetkellä tekemässä. Keskeytyksen lauetessa muodostuu tapahtumaa kuvaava viesti. Käsittelysekvenssi alkaa keskeytyksestä ja käsittää yhden tai useampia käsittelyitä peräkkäin. Prioriteetti tarkoittaa käsittelyn kiireellisyydestä. Mitä korkeampi
- 25 prioriteetti on, sitä kiireellisempi käsittely on kyseessä. Skedulointi tarkoittaa sellaista käsittelijän toimintaa, jolla ratkaistaan, minkä taskin suoritusta on aloitettava tai jatkettava.

- Keksinnön mukainen parannettu menetelmä taskien
- 30 käsittelemiseksi voidaan toteuttaa keskeyttävänä prioriteettiskedulointina, joka itse on keskeytettävissä. Seuraavassa selostetaan ensin prioriteettiskedulointiin perustuva menetelmä ja tämän jälkeen määräaikaskedulointiin perustuva menetelmä.

- 35 Keksinnön mukaisessa parannetussa menetelmässä taskien käsittelemiseksi taskit on varustettu prioriteeteilla ja taskit käsitellään siten, että taski suorite-

taan aina loppuun siten, että vain korkeamman prioriteetin omaavan taskin suoritus voi sen suoritusta hidastaa. Näin ollen eri prioriteetin omaavien taskien käsitteleminen toteutetaan reaaliajassa erityisesti sulautetuissa järjestelmissä aina prioriteettijärjestyksessä ja ensimmäistä niistä taskeista, joilla on työtä. Tätä kutsutaan prioriteettiskeduloinniksi. Keksinnön mukaisesti aina, kun syntyy uusi viesti, joko tapahtuman seurauksena tai taskin tuottamana, tarkastetaan sitä

10 käsittelevän taskin prioriteetti ja viesti talletetaan käsittelemättömien viestien joukkoon prioriteettitiedon kera. Tämän jälkeen tarkastetaan, onko korkein käsitte-lyä odottavan viestin prioriteetti korkeampi kuin suori-tettavan taskin A prioriteetti. Jos on, annetaan tämän

15 palveleminen työksi sille taskille B, jolle se kuuluu ja taskista B tehdään suoritettava. Entinen suoritettava taski A merkitään odottavaksi, ja ruvetaan suorittamaan taskia B. Jos korkeampaa prioriteettia ei löydy, jat-ketaan suoritettavaa taskia A. Kun suoritettava taski X

20 saa työnsä valmiiksi, etsitään korkeaprioriteettisin odottava taski Y ja merkitään se suoritettavaksi. Sitten käsitellään taskin X tuottamat viestit normaaliin tapaan uusina viesteinä.

Keksinnön mukaisessa menetelmässä aina kun

25 havaitaan tapahtuma, aloitetaan sen käsittely heti, vaikka edellisen tapahtuman käsittely olisi kesken, ellei sitä ole erityisesti kielletty. Seuraava kuvaus esittää yhtä tällaista käsittelysekvenssiä. Tapahtuma aiheuttaa keskeytyksen, josta syntyy aina tapahtumaa

30 kuvaava viesti ja uusi käsittelysekvenssi heti, kun käynnissä olevassa käsittelysekvenssissä on keskeyt-täminen sallittu. Näin yhdellä käsittelijällä voi olla useita käsittelysekvenssejä kesken viimeksi aloitetun käsittelysekvenssin ollessa käynnissä. Menetelmän mukai-

35 sessa käsittelysekvenssissä

1) kielletään uusien tapahtumien käsittely; määrätään viestiä käsittelevä taski eli ensimmäinen taski; ensim-

- mäisen taskin tietoihin merkitään, että ensimmäisen taskin on käsiteltävä uusi viesti; ensimmäisen taskin prioriteetti tarkistetaan ja, mikäli se on korkeampi kuin juuri vallitseva prioriteetti eli suoritettavana olevan toisen taskin prioriteetti, suoritetaan seuraavat toimenpiteet 11 - 12;
- 11) talletetaan vallitseva prioriteetti käsittelykohtaiseksi tiedoksi vanha-prioriteetti; vallitsevaksi prioriteetiksi asetetaan ensimmäisen taskin prioriteetti;
- 12) niin kauan kuin vallitseva prioriteetti on korkeampi kuin vanha-prioriteetti, toteutetaan seuraava menetelmävaihe 121;
- 121) jos on olemassa kolmas taski, jonka prioriteetti on vallitseva prioriteetti ja jolla on käsiteltävänä viesti, suoritetaan toimenpiteet 1211 - 1212, muuten vallitsevaksi prioriteetiksi vaihdetaan lähinnä alempi prioriteetti;
- 1211) sallitaan uusien tapahtumien käsittely; edellisessä kohdassa löydetty kolmas taski suoritetaan; suorituksesta voi syntyä viestejä, jotka on taskien käsiteltävä; 1212) kustakin syntyneestä viestistä määrätään sitä käsittelevä taski eli neljäs taski; neljännen taskin tietoihin merkitään, että neljännen taskin on käsiteltävä syntynyt viesti; neljännen taskin prioriteetti tarkistetaan ja, mikäli se on korkeampi, kuin juuri vallitseva prioriteetti, vaihdetaan vallitsevaksi prioriteetiksi neljännen taskin prioriteetti; ja kielletään uusien ulkoisten tapahtumien käsittely;
- 2) sallitaan uusien tapahtumien käsittely ja jatketaan keskeytynyttä toimintaa.

Keksinnön mukaisen menetelmän eräässä sovellutuksessa suoritusta odottavien taskien työtilat järjestetään yhteen pinomuistiin, jossa suorituksessa olevan taskin työtila on aina päällimmäisenä.

Keksinnön mukaisessa menetelmässä taskit eivät koskaan kesken suorituksensa odota mitään. Tästä seuraa,

että suorituksen aikainen työtila voidaan varata yhte-
näisestä muistialueesta pinomaisesti järjestyksessä
siten, että koko ajan on olemassa paikka, pinon huippu,
jonka toisella puolella (alapuoli) on käytössä olevaa
5 muistitilaa ja toisella puolella on vapaata muistitilaa.
Suorituksensa aikana taski varaa ja vapauttaa työtilaa.
Kun taskin suoritus loppuu, se vapauttaa kaiken työ-
tilansa. Prioriteettiskedulointisäännöistä seuraa: jos
korkeampi prioriteettisen taskin suoritus on kesken, ei
10 koskaan suoriteta matalampi prioriteettista taskia.
Suorituksessa olevan taskin työtila on aina pinon pää-
limmäisenä, ja keskeytettyjen taskien työtilat ovat
siitä alaspäin prioriteettijärjestyksessä ilman tyhjiä
välejä.

15 Keksinnön etuna on, että siinä yhdistetään
keskeytys- ja monitaskimenetelmän parhaat puolet.

 Monitaskimenetelmän mukaisesti yhden käsitteli-
jän toiminnot jaetaan kiireellisyys- eli prioriteet-
tiluokkiin, joista muodostetaan eri prioriteetin omaavia
20 taskeja. Taskit sisältävät tietorakenteen käsiteltäviä
tietoja varten ja käsittelyohjeiston kuvaamaan varsi-
naista käsittelyä. Taskit voivat kommunikoida toistensa
ja ympäristön kanssa. Näin samasta ulkoisesta tapah-
tumasta voidaan hallitusti tuottaa eri kiireellisyysas-
25 teisia vasteita.

 Keskeytysmenetelmän mukaisesti taskit ovat
tapahtuman käsittelijöitä. Taskille tuleva tapahtuma
voi olla laitetapahtuma tai viestin tuleminen toisesta
tai samasta taskista. Koska taskin suoritus voi kes-
30 keytyä vain korkeamman prioriteetin taskin suorituksen
ajaksi, ja vastaavasti taskia suoritettaessa ei ole
korkeamman prioriteetin taskeja kesken, voidaan taskien
työmuistitilana käyttää yhtä yhteistä pinomaisesti
varattavaa muistialuetta. Toisaalta tällaiset taskit
35 voidaan toteuttaa alistettuina tapahtumakäsittelyrutiini-
neina, jotka ovat täysin funktionaalisia, eli niiden
tulokset riippuvat pelkästään syöttötiedoista. Tällais-

ten funktionaalisten käsittelyrutiinien testaaminen on vakiintunutta tekniikkaa eikä vaadi erikoisapuneuvoja.

Keksinnön etuna on edelleen, että se on yksinkertainen ja hierarkkinen menetelmä, joka on helposti sovitettavissa sulautettuihin järjestelmiin. Parannettu menetelmä taskien käsittelemiseksi tarjoaa erillisen käyttöympäristön, jonka avulla yksinkertaisesti ja tehokkaasti voidaan kutsua ja käynnistää eri taskeja. Koska myös seuraavan suoritettavan taskin etsiminen (skedulointityö) on keskeytettävissä, menetelmä käy suurien taskimäärien tehokkaaseen skedulointiin.

Edelleen keksinnön etuna on, että se on suhteellisen helposti sovellettavissa erilaisiin ympäristöihin. Sen avulla voidaan ohjata ja järjestää elektronisten tai mekaanisten laitteitten toimintoja ja/tai tietokoneohjelmien suorittamista.

Elektronisen laitteen ollessa kyseessä keksinnön mukainen menetelmä voidaan integroida osaksi laitetta, esim. osaksi tietokoneen keskusyksikön sisäisiä piirejä, jolloin tietokone voi ohjata esim. ohjelmallisesti toteutettuja taskeja ilman erillistä käyttöjärjestelmää. Tästä olisi huomattavaa etua tehokkuuden kannalta. Integrointi voidaan toteuttaa esimerkiksi sisällyttämällä mikroprosessoriin keksinnön mukaista menetelmää toteuttava osa tai valmistamalla ns. oheispiiri, joka liitettynä tietokoneen elektroniikkaan ohjaa taskeja esitettyä menetelmää käyttäen.

Keksinnön mukainen menetelmä voidaan sijoittaa ohjelmistototeutuksena käyttöjärjestelmään. Viestit ja tietorakenteet voidaan toteuttaa muistipaikkojen ja aliohjelmakutsujen avulla. Näin järjestetty tietokone on laite, joka toimii esitetyn menetelmän mukaisesti.

Keksinnön mukaisessa parannetussa menetelmässä taskien käsittelmiseksi reaaliajassa erityisesti sulautetuissa järjestelmissä taskit on varustettu palveluajoilla ja taskeista kukin suoritetaan aina loppuun siten, että vain sellaisen taskin, jolla on aikaisempi

määräaika, suoritus voi sitä hidastaa. Tätä kutsutaan määräaikaskeduloinniksi. Keksinnön mukaisesti aina kun havaitaan tapahtuma, aloitetaan sen käsittely heti, vaikka edellisen tapahtuman käsittely olisi kesken,
5 ellei sitä ole erityisesti kielletty. Seuraava selostus esittää yhtä käsittelysekvenssiä. Tapahtuma aiheuttaa keskeytyksen, josta syntyy aina tapahtumaa kuvaava viesti ja uusi käsittelysekvenssi heti, kun käynnissä olevassa käsittelysekvenssissä on keskeyttäminen sallit-
10 tu. Näin yhdellä käsittelijällä voi olla useita käsittelysekvenssejä kesken viimeksi aloitetun käsittelysekvenssin ollessa käynnissä. Käsittelysekvenssi sisältää seuraavat vaiheet:

1) kielletään uusien tapahtumien käsittely; määrätään
15 viestiä käsittelevä taski eli ensimmäinen taski; ensimmäisen taskin tietoihin merkitään, että ensimmäisen taskin on käsiteltävä uusi viesti; ensimmäisen taskin palveluaika ja kellosta katsottu nykyhetki lasketaan yhteen, ja saatu tulos talletetaan tapahtuman määräajak-
20 si. Mikäli uuden tapahtuman määräaika on aikaisempi kuin vallitseva määräaika eli suoritettavana olevan toisen taskin käsiteltävänä olevan tapahtuman määräaika, suoritetaan seuraavat toimenpiteet 11 - 12;

11) talletetaan vallitseva määräaika käsittelykohtaisek-
25 si tiedoksi vanha-määräaika; vallitsevaksi määräajaksi asetetaan uuden tapahtuman määräaika; sallitaan uusien ulkoisten tapahtumien käsittely;

12) niin kauan kuin vallitseva määräaika on aikaisempi kuin vanha-määräaika, toistetaan seuraavaa menetelmä-
30 vaihetta 121;

121) jos on olemassa kolmas taski, jolla on käsiteltävänä viesti, jonka määräaika on vallitseva määräaika, suoritetaan toimenpiteet 1211 - 1212, muuten etsitään käsittelemätön viesti, joka on määräaikajärjestyksessä
35 seuraava, ja asetetaan sen määräaika vallitsevaksi määräajaksi;

1211) sallitaan uusien tapahtumien käsittely; edellis-

sä kohdassa löydetty kolmas taski suoritetaan; suorituksesta voi syntyä viestejä, jotka on taskien käsiteltävä; 1212) katsotaan kellosta nykyhetki; kustakin syntyneestä viestistä määrätään tapahtumaa käsittelevä taski eli
 5 neljäs taski, neljännen taskin tietoihin merkitään, että neljännen taskin on käsiteltävä syntynyt viesti, neljännen taskin palveluaika tarkistetaan lisätään se nykyhetkeen ja asetetaan summa tapahtuman määräajaksi ja, mikäli se on aikaisempi, kuin juuri vallitseva
 10 määräaika, vaihdetaan se vallitsevaksi määräajaksi; kielletään uusien ulkoisten tapahtumien käsittely;
 2) sallitaan uusien tapahtumien käsittely ja jatketaan keskeytynyttä toimintaa.

Puhtaassa prioriteettiskeduloinnissa prioriteetit on kiinnitetty pysyvästi taskeihin. On osoitettu,
 15 että käsittelijän (prosessorin) kapasiteettia voidaan eräissä tapauksissa käyttää paremmin hyväksi, kun prioriteetit määrätään vasta käsiteltävän tapahtuman hetkellä. Määräaikaskeduloinnissa suoritetaan aina sitä
 20 taskia, jolla on aikaisin vaadittu valmistumismääräaika.

Määräaikaskedulointi, jolla on samat edut kuin esitellyllä prioriteettiskeduloinnilla, voidaan toteuttaa tekemällä seuraavat muutokset edellä esitettyyn keksinnön mukaiseen prioriteettiskedulointiin:

- 25 a) Kiinnitetään jokaiseen taskiin palveluaika, mikä on pisin sallittu aika käsiteltävän viestin syntymisestä taskin valmistumiseen. Tämä tulee prioriteetin tilalle.
- b) Kun syntyy viesti, joka on taskin käsiteltävä, katsotaan kellosta nykyhetki, lisätään taskin palveluaika,
 30 ja asetetaan summa viestin määräajaksi.
- c) Pidetään kirjaa vallitsevan ja vanhan prioriteetin asemesta vallitsevasta ja vanhasta määräajasta.
- d) Suoritukseen valitaan aina taski, jolla on käsiteltävänä viesti, jolla on aikaisin määräaika.

35 Keksinnön mukaista parannettua menetelmää taskien käsittelemiseksi voidaan soveltaa erilaisissa tietokonejärjestelmissä ja erityisesti sulautetuissa

järjestelmissä, joihin kuuluu esim. mekaaninen laitekokonaisuus ja tätä ohjaava tietokonejärjestelmä. Tällaisessa järjestelmässä keksinnön mukaisella menetelmällä saavutetaan mm. seuraavia etuja:

- 5 Useimmat nykyaikaiset mikroprosessorit toteuttavat aliohjelmakutsut pinon avulla. Tällaiselle prosessorilla riittää uutta menetelmää käytettäessä yksi yhteinen pino kaikille taskeille, koska tapahtumien tultua käsitellyiksi pinoa voidaan purkaa tyhjäksi asti.
- 10 Vaikka pinoon tarvitaan tilaa samoille toiminnoille ja keskeytyksille kuin useita pinoja käytettäessä, muistitilaa tarvitaan vähemmän. Tämä siksi, että keskeytyksille ei tarvitse varata tilaa useasta pinosta.

- Keksinnön mukaisen parannetun menetelmän mukaan
- 15 tehty reaaliaikainen eri prioriteetin omaavien taskien suoritusmenetelmä on huomattavasti nopeampi ja käyttää vähemmän muistia kuin aikaisemmat tunnetut järjestelmät. Parannettu taskien käsittelymenetelmä on tiiviimpi järjestelmävaiheeltaan ja sen suorituskapasiteetti on
- 20 huomattavasti parempi kuin aikaisemmilla järjestelmillä.

- Keksinnön mukaisen parannetun taskien käsittelymenetelmän etuja voidaan käyttää hyväksi ainakin kolmella eri tavalla. Ensiksikin voidaan toteuttaa uusi laitteistokokonaisuus, jolla on nykyisen laitteiston
- 25 suorituskkyky, halvemmissa osista, koska tarvitaan vähemmän muistipiirejä ja prosessorina voidaan käyttää hitaampaa mallia. Toiseksi voidaan toteuttaa uusia laitteistoja, joilla on parempi suorituskkyky kuin nykyisillä laitteistoilla. Kolmanneksi täysin uusien laitteistojen
- 30 suunnittelu helpottuu, koska prosessorin suorituskkyky riittää paremmin silloinkin, kun tehtävien kiireellisyys- ja suoritusjärjestystä ei ole ratkaistu parhaalla mahdollisella tavalla.

- Seuraavassa keksintöä, sen erästä sovellusympäristöä, sen toimintaa ja sen muita etuja selostetaan oheisten piirustusten avulla, joissa
- 35 kuva 1 esittää kaaviomaisesti kerrostalon hissiä;

kuva 2 esittää lohkokaaavion muodossa hissiä ja sen ohjausjärjestelmää;

kuva 3 esittää tietovirtakaaviona hissin ohjausjärjestelmän liittymistä hissiin;

5 kuva 4 esittää tietovirtakaaviona hissin ohjausjärjestelmän taskeja ja niiden välisiä viestiyhteyksiä; ja

kuva 5 esittää hissin eri taskien suoritusta aikatasossa; ja

kuva 6 esittää pinomaisen työmuistin käyttöä kuvan 5
10 taskeja suoritettaessa.

Kuvassa 1 on kaaviomaisesti esitetty rakennus 10, johon on järjestetty hissi ja sen ohjausjärjestelmä. Hissikori 3 on sovitettu liikkumaan kerrosten K1, K2, K3, K4 ja K5 välillä hissikuilussa 1. Hissikori 3 on
15 ripustettu kuiluun 1 köysien 2a avulla ja sitä siirretään nostomoottorilla 2 kuilussa 1. Kullakin kerrostanteella on ensimmäiset kerrospainikkeet 1a; P1, P2, P3, P4 ja P5 hissin kutsumista varten. Hissikorissa 3 on toiset kerrospainikkeet 3a; H1, H2, H3, H4 ja H5.
20 Hissikorin oveen 4 tai yleensä ovijärjestelyyn kuuluu ovimoottori 4a (vrt. kuva 2) oven 4 avaamiseksi, ovisilmä 4b oven välissä olevien esteiden, kuten ihmisten havaitsemiseksi ja kytkimet (ei esitetty piirustuksessa) oven asennon havaitsemiseksi.

25 Hissikorin 3, kerrosten ja nostolaitteiden ohjaus- ja toimilaitteet, kuten kerrospainikkeet 1a, 3a ja nostomoottori 2, on yhdistetty liitäntäelektronikan 5 välityksellä ohjaustietokoneeseen 6. Ohjaustietokone on lisäksi yhdistetty syöttö- ja tulostuslaitteisiin 7.

30 Kuvassa 2 hissin ohjausjärjestelmä on esitetty lohkokaaaviona. Ohjaustietokoneeseen 6 kuuluu keskusyksikkö 11 ja apuyksikkö 12. Apuyksikköön 12 kuuluu vuorostaan käyttöjärjestelmäyksikkö 13 ja sovellusyksikkö 14. Keksinnön mukainen parannettu menetelmä taskien
35 käsittelemiseksi on olennainen osa käyttöjärjestelmäyksikköä 13. Se voidaan toteuttaa ohjelmallisesti tai se voi olla osittain ohjelmallinen ja osittain liittyä

fyysisiin rakenteisiin. Käyttöjärjestelmäyksikkö valitsee käynnistettävän taskin ja huolehtii taskeille menevistä viesteistä.

Syöttö- ja tulostuslaitteisiin 7 kuuluu näyttö 5 8 ja näppäimistö 9. Näyttöjä on useita ja ne voivat sijaita esim. eri kerrostasoilla K1, K2,... ja lisäksi etäisyyden päässä valvontahuoneessa tai vastaavassa. Järjestelmään on lisäksi yhdistetty teholähde 21, jonka kautta kaikki sähköiset laitteet hissijärjestelmässä 10 saavat tehonsa.

Kuvassa 3 on esitetty tietovirtakaavion muodossa sovellusyksikkö 14 ja sen sisältämien sovellutusten yhteydet ja liitännät syöttö- ja tulostuslaitteisiin 7 sekä liitäntäelektronikan 5 kautta hissikuilun 1 ja 15 hissikorin 3 laitteisiin. Nämä liitännät voidaan jaotella seuraavasti: matkustaja-hissikorissa 15 (Passenger-in-Car), matkustaja-ovella 16 (Passenger-in-Door), matkustaja-kerrostasanteella 17 (Passenger-at-Level), moottorinohjaus 18 (Car-Drive-Motor), kerrosilmaisimet 20 19 (Floor-Sensors) ja hissikorin-ovi 20 (Car-Doors). Liitäntöihin liittyvät tapahtumat on merkitty katkovii-voin ja niitä kuvaavin nimityksin kuvaan 3.

Tapahtumat jotka liittyvät hissikorissa olevaan matkustajaan eli matkustaja-hissikorissa 15 (Passenger-in-Car) liitäntään tulevat sovellusyksikölle 14 keskeytyksinä aja-kerrokseen-1 ... aja-kerrokseen-5 (Drive-to-Floor-1 ... Drive-to-Floor-5) sekä pysäytä-ajo (Stop-Driving) hissikorin 3 painikkeiden 1a (vrt. kuva 1) kautta. Tapahtumat, jotka liittyvät ovelle olevaan 30 matkustajaan eli matkustaja-ovella 16 (Passenger-in-Door) liitäntään tulevat sovellusyksikölle 14 keskeytyksenä joku-ovella (Somebody-in-Door) jne. siten, että kun matkustaja neljännessä kerroksessa painaa kutsunappia P4, tapahtuu keskeytys, ja tieto napin painalluksesta 35 on sovelluksen käytettävissä tai, kun hissi saapuu kerrosalueelle tai poistuu siitä, tapahtuu keskeytys ja tilannetta kuvaava tieto on sovellusyksikön käytettävissä.

sä. Vastaavalla tavalla voidaan selostaa muut kuvassa 3 esitetyt liitännät ja niihin liittyvät tapahtumat merkittyihin englanninkielisiin nimityksiin viitaten.

Kuvassa 4 on esitetty sovellusyksikön 14 jakaantuminen taskeihin. Taskien välillä olevat viestiyhteydet on esitetty sekä katkoviivoin että ehyinä viivoina. Olennainen osa hissin ohjaukseen liittyvistä tapahtumista on merkitty kuvan viestiyhteyksiin englanninkielisin nimityksin. Erityisen kiireelliset toiminnot on kerätty taskiin ohjausliitäntä eli control-interface 14a. Näitä kiireellisiä toimintoja ovat ulkoisiin tapahtumiin liittyvien tietojen sieppaaminen ennen niiden häviämistä ja ohjaukseen liittyvien tietojen siirtäminen toimilaitteille. Ohjauslogiikka on keskitetty taskiin hissin-käyttö eli Drive-Lift 14b ja näyttöjen ohjaukseen liittyvä toiminta, joka ei ole inhimillisen reagoinnin hitauden vuoksi kovin kiireellistä, taskiin näytön-ohjain eli display-driver 14c. Taskien prioriteetit on asetettu siten, että taskilla control-interface 14a on korkein prioriteetti ja taskilla display-driver 14c on matalin prioriteetti. Kun taski lähettää viestin toiselle, se samalla aiheuttaa vastaanottavan taskin käynnistymisen heti tai myöhemmin. Esimerkiksi, kun taski control-interface 14a on käsitellyt pysäytä-ajo (Stop-Driving) keskeytyksen, se lähettää hätäpysäytys (Emergency-Stop) viestin taskille Drive-Lift 14b, mikä aiheuttaa viimeksimainitun käynnistymisen. Käyttöjärjestelmä huolehtii taskien suoritusjärjestyksestä ja niiden käynnistämisestä.

Kuvassa 5 on esitetty esimerkkisekvenssi, joka havainnollistaa keksinnön mukaisen taskien käsittelymenetelmän toimintaa hissinohjauksessa. Käyttöjärjestelmäyksikkö 13 suorittaa menetelmän mukaisesti taskien skeduloinnin.

Kuva 6 esittää pinomaisen työmuistin käyttöä kuvan 5 esimerkissä. Käytetyn muistitilan koko on V ja sitä seurataan ajan t suhteen. Eri taskien tarvitsemt

työtilat on merkitty erilaisin viivoituksin, joiden merkitys on esitetty kuvassa 5.

Lähtötilanteessa hissikoria 3 siirretään ylöspäin kohti toista kerrosta kuvan 1 kerrostalossa; toiseen kerrokseen ei ole kutsua voimassa, mutta tätä ylempään kerrokseen on. Ohjausjärjestelmällä ei ole mitään tekemistä, joten suoritetaan ns. joutotaskia Idle-Loop. Joutotaski on aina olemassa, mutta sitä ei esitetä kaavioissa. Vallitseva prioriteetti on joutotaskin prioriteetti 0. Nimetään esittelyssä käsittelysekvenssit, jotka voivat keskeyttää toisensa, kirjaimilla a, b,... Joutotaskin käsittelysekvenssi on a ja muut käsittelysekvenssit alkavat keskeytyksistä ja päättyvät, kun palataan suorittamaan viimeksi keskeytynyttä käsittelysekvenssiä. Joutotaskia suoritettaessa työmuistissa on vain joutotaskin tarvitsema työtila.

Hissikori 3 saapuu toisen kerroksen alueelle hetkellä t1 ja aiheuttaa keskeytyksen hissikori-eikerrostasolla (Car-not-at-Floor) tilan loppuessa. Keskeytyksestä syntyy käsittelysekvenssi b. Control-interface taski 14a käsittelee tästä keskeytyksestä syntyvän viestin. Koska taskin control-interface prioriteetti 3 on korkeampi kuin vallitseva prioriteetti 0, jätetään Idle-Loop taski keskeytetyksi ja ryhdytään suorittamaan taskia control-interface. Käsittelysekvenssin b vanha-prioriteetti on entinen vallitseva prioriteetti 0, ja vallitseva prioriteetti on nyt 3. Työmuistissa on control-interface taskin työtila joutotaskin työtilan päällä, kuten kuvassa 6 on esitetty.

Control-interface taski 14a saa tehtävänsä loppuun hetkellä t2 ja se lähettää viestin kerrostasolla (At-Floor) tilanteen syntymisestä. Drive-Lift taski 14b käsittelee tämän viestin. Koska prioriteetilla 3 ei löydy enää suoritusta kaipaavia taskeja, vaihdetaan menetelmän mukaisesti vallitsevaksi prioriteetiksi tätä alempi prioriteetti 2 ja löydetään taski Drive-Lift 14b, joka kaipaa suoritusta; ryhdytään suorittamaan taskia

Drive-Lift. Nyt työmuistissa Drive-Lift taskin työtila joutotaskin työtilan päällä (kuva 6).

Drive-Lift taski 14b saa työnsä valmiiksi hetkellä t3. Koska hissi ei pysähdy tähän kerrokseen, 5 ainoa asia, mitä tästä kaikesta seuraa, on näytettävän kerrosnumeron muuttaminen kakkosesta kolmoseksi. Drive-Lift taski siis lähettää näytä-kerrosto (Indicated-Floor) viestinä uuden kerrosnumeron display-driver taskille 14c. Muita viestejä ei lähetetä. Seurauksena 10 vallitsevaksi prioriteetiksi vaihdetaan 1 ja ryhdytään suorittamaan taskia display-driver. Nyt työmuistissa on display-driver taskin työtila joutotaskin työtilan päällä (kuva 6).

Matkustaja painaa hätäpysäytysnappia hetkellä 15 t4 ja aiheuttaa pysäytä-ajo (Stop-Driving) keskeytyksen. Keskeytyksestä syntyy käsittelysekvenssi c. Koska syntyvää viestiä käsittelee prioriteetin 3 omaava taski control-interface 14a ja vallitseva prioriteetti on vain 1, jätetään myös display-driver taski 14c keskeyte- 20 tyksi ja ryhdytään suorittamaan control-interface taskia 14a. Vallitseva prioriteetti on tästä alkaen 3 ja käsittelysekvenssin c vanha-prioriteetti on 1. Nyt työmuistissa on control-interface taskin 14a työtila display-driver taskin 14c ja joutotaskin työtilan pääl- 25 lä (kuva 6).

Control-interface taski valmistuu hetkellä t5 ja lähettää hätäpysäytys (Emergency-Stop) viestin Drive-Lift taskille 14b. Koska prioriteetin 3 omaavia taskeja ei enää ole suoritusta odottamassa, siirrytään suorittamaan Drive-Lift taskia 14b ja asetetaan vallitsevaksi 30 prioriteetiksi 2. Nyt työmuistissa on Drive-Lift taskin työtila display-dirver taskin ja joutotaskin työtilan päällä (kuva 6).

Drive-Lift taskin 14b suorituksen lopputulos 35 selviää hetkellä t6, jonka mukaan hissikori 3 on pysäytettävä, ja pysäytys- (Stop) viesti lähetetään control-interface taskille 14a, sekä samalla osoita-kerros

(Indicated-Floor) viestinä kerrosnumero kaksi taskille display-driver 14c. Jälkimmäinen viesti tarvitaan, koska hissikori 3 on pysäytettävä ajamatta kerrokseen kolme. Koska control-interface taskin prioriteetti on vallitsevaa prioriteettia korkeampi, nostetaan vallitseva prioriteetti vastaanottajan tasolle eli 3:ksi. Prioriteettitasolla 3 on suoritusta odottava taski control-interface, joka saman tien käynnistetään. Nyt työmuistissa on control-driver taskin työtila display-driver taskin ja joutotaskin työtilan päällä (kuva 6).

Control-interface taski 14a asettaa laitteisto-ohjaukset siten, että hissi alkaa jarruttaa. Taskin suoritus loppuu hetkellä t7. Nyt menetelmän mukaan etsitään suoritettavia taskeja vallitsevalta prioriteettitasolta 3. Ei löydy. Vaihdetaan vallitsevaksi prioriteettitasoksi 2, eikä sieltäkään löydy. Vaihdetaan vallitsevaksi prioriteettitasoksi 1, ja nyt huomataan, että se ei ole enää suurempi kuin käynnissä olevan käsittelysekvenssin c vanha prioriteetti. Tästä syystä käsittelysekvenssi c loppuu tähän ja jatketaan käsittelysekvenssiä b ja display-driver taskia. Nyt työmuistissa on display-driver'in työtila joutotaskin työtilan päällä (kuva 6).

Display-driver taski 14c on saanut kerrosnäytön asetettua kolmoseksi hetkellä t8 ja lopettaa. Prioriteettitasolta 1 löytyy käsittelemätön tapahtuma: taskille display-driver on lähetetty osoita-kerros (Indicated-Floor) viestissä kerrosnumero kaksi. Seurauksena ryhdytään suorittamaan uudelleen taskia display-driver. Nyt työmuistissa on taas display-driver'in työtila joutotaskin työtilan päällä.

Display-driver taski on saanut kerrosnäytön asetettua kakkoseksi hetkellä t9 ja lopettaa. Muita suoritettavia taskeja ei enää löydy vallitsevalta prioriteettitasolta 1, ja vallitsevaksi prioriteetiksi valitaan 0. Tämä ei ole enää suurempi kuin käynnissä olevan käsittelysekvenssin b vanha prioriteetti. Tästä

syystä käsittelysekvenssi b loppuu tähän ja jatketaan käsittelysekvenssiä a ja joutotaskia. Nyt työmuistissa on vain joutotaskin työtila (kuva 6).

Keksintöä ei rajata pelkästään hissinohjausta
5 koskevaksi. Hissinohjaus on esimerkkinä hyvä sovellus-
kohde parannetulle taskien käsittelymenetelmälle, joka
on täysin yleiskäyttöinen, kun tarvitaan reaaliajassa
eri kiireysasteen vasteiden tuottamista yhdellä käsit-
telijällä. Monet muunnokset ovat mahdollisia pysyttäessä
10 patenttivaatimusten määrittelemän keksinnöllisen ajatuk-
sen puitteissa.

PATENTTIVAATIMUKSET

1. Parannettu menetelmä taskien käsittelyksi reaaliajassa erityisesti sulautetuissa järjestelmissä, jossa taskit on varustettu prioriteeteilla, ja joista taskeista kukin suoritetaan aina loppuun siten, että vain korkeampiprioriteettisen taskin suoritus voi sen suoritusta hidastaa, jossa menetelmässä aina kun havaitaan tapahtuma, sen käsittely aloitetaan menetelmän mukaisesti heti, vaikka edellisen tapahtuman käsittely olisi kesken, ellei sitä ole erityisesti kielletty, jolloin yhdellä käsittelijällä voi olla useita käsittelysekvenssejä kesken viimeksi aloitetun käsittelysekvenssin ollessa käynnissä, jossa menetelmässä ulkoinen tapahtuma aiheuttaa keskeytyksen, josta syntyy tapahtumaa kuvaava viesti ja uusi käsittelysekvenssi heti, kun käynnissä olevassa käsittelysekvenssissä on keskeyttäminen sallittu, t u n n e t t u siitä, että heti keskeytyksen tapahduttua syntyvässä käsittelysekvenssissä:
- 1) kielletään uusien tapahtumien käsittely, määrätään keskeytyksestä syntynyttä viestiä käsittelevä ensimmäinen taski, ensimmäisen taskin tietoihin merkitään, että ensimmäisen taskin on käsiteltävä uusi viesti, ensimmäisen taskin prioriteetti tarkistetaan ja, mikäli se on korkeampi kuin juuri vallitseva prioriteetti eli suoritettavana olevan toisen taskin prioriteetti, suoritetaan seuraavat toimenpiteet 11 - 12;
 - 11) talletetaan vallitseva prioriteetti käsittelykohtaiseksi tiedoksi vanha-prioriteetti; vallitsevaksi prioriteetiksi asetetaan ensimmäisen taskin prioriteetti; sallitaan uusien tapahtumien käsittely;
 - 12) niin kauan kuin vallitseva prioriteetti on korkeampi kuin vanha-prioriteetti, toteutetaan seuraava menetelmävaihe 121;
 - 121) kielletään uusien tapahtumien käsittely; jos on olemassa kolmas taski, jonka prioriteetti on vallitseva

prioriteetti ja jolla on käsiteltävänä viesti, sallitaan uusien tapahtumien käsittely ja suoritetaan toimenpiteet 1211 - 1212, muuten vallitsevaksi prioriteetiksi vaihdetaan lähinnä alempi prioriteetti; ja sallitaan uusien tapahtumien käsittely;

1211) edellisessä kohdassa löydetty kolmas taski suoritetaan; suorituksesta voi syntyä viestejä, jotka on taskien käsiteltävä;

1212) kustakin syntyneestä viestistä määrätään sitä käsittelevä neljäs taski, neljännen taskin tietoihin merkitään, että neljännen taskin on käsiteltävä syntynyt viesti, neljännen taskin prioriteetti tarkistetaan ja, mikäli se on korkeampi kuin juuri vallitseva prioriteetti, vaihdetaan vallitsevaksi prioriteetiksi neljännen taskin prioriteetti;

2) sallitaan uusien tapahtumien käsittely ja jatketaan keskeytynyttä toimintaa.

2. Parannettu menetelmä taskien käsittelemiseksi reaaliajassa erityisesti sulautetuissa järjestelmissä, jossa taskit on varustettu palveluajoilla, ja joista taskeista kukin suoritetaan aina loppuun siten, että vain sellaisen taskin, jolla on aikaisempi määräaika, suoritus voi sitä hidastaa, jossa menetelmässä aina kun havaitaan tapahtuma, aloitetaan sen käsittely heti tällä menetelmällä, vaikka edellisen tapahtuman käsittely olisi kesken, ellei sitä ole erityisesti kielletty, jolloin yhdellä käsittelijällä voi olla useita käsittelysekvenssejä kesken viimeksi aloitetun käsittelysekvenssin ollessa käynnissä, ja jossa menetelmässä ulkoinen tapahtuma aiheuttaa keskeytyksen, josta syntyy aina tapahtumaa kuvaava viesti ja uusi käsittelysekvenssi heti, kun käynnissä olevassa käsittelysekvenssissä on keskeyttäminen sallittu, t u n n e t t u siitä, että heti keskeytyksen tapahduttua syntyvässä käsittelysekvenssissä:

1) kielletään uusien tapahtumien käsittely; määrätään viestiä käsittelevä ensimmäinen taski; ensimmäisen

- taskin tietoihin merkitään, että ensimmäisen taskin on käsiteltävä uusi viesti; ensimmäisen taskin palveluaika ja kellosta katsottu nykyhetki lasketaan yhteen, ja saatu tulos talletetaan tapahtuman määrääjäksi; mikäli
- 5 uuden tapahtuman määräaika on aikaisempi kuin vallitseva määräaika eli suoritettavana olevan toisen taskin käsiteltävänä olevan tapahtuman määräaika, suoritetaan seuraavat toimenpiteet 11 - 12;
- 11) talletetaan vallitseva määräaika käsittelykohtaiseksi tiedoksi vanha-määräaika; vallitsevaksi määrääjäksi asetetaan uuden tapahtuman määräaika; sallitaan uusien ulkoisten tapahtumien käsittely;
- 10 12) niin kauan kuin vallitseva määräaika on aikaisempi kuin vanha-määräaika, toistetaan seuraavaa menetelmä-
- 15 vaihetta 121;
- 121) kielletään uusien tapahtumien käsittely; jos on olemassa kolmas taski, jolla on käsiteltävänä viesti, jonka määräaika on vallitseva määräaika, sallitaan uusien tapahtumien käsittely ja suoritetaan toimenpiteet 1211 - 1212, muuten etsitään käsittelemätön viesti,
- 20 joka on määräaikajärjestyksessä seuraava, ja asetetaan sen määräaika vallitsevaksi määrääjäksi ja sallitaan uusien tapahtumien käsittely,
- 1211) edellisessä kohdassa löydetty kolmas taski suoritetaan; suorituksesta voi syntyä viestejä, jotka on taskien käsiteltävä;
- 25 1212) katsotaan kellosta nykyhetki; kustakin syntyneestä viestistä määrätään tapahtumaa käsittelevä neljäs taski, neljännen taskin tietoihin merkitään, että neljännen taskin on käsiteltävä syntynyt viesti, neljännen taskin palveluaika tarkistetaan ja lisätään se nykyhetkeen ja asetetaan summa tapahtuman määrääjäksi ja, mikäli se on aikaisempi kuin juuri vallitseva määräaika, vaihdetaan se vallitsevaksi määrääjäksi;
- 30 2) sallitaan uusien tapahtumien käsittely ja jatketaan keskeytynyttä toimintaa.
- 35

3. Patenttivaatimusten 1 tai 2 mukainen mene-

telmä, t u n n e t t u siitä, että suoritusta odottavien taskien työtilat järjestetään yhteen pinomui-
tiin, jossa suorituksessa olevan taskin työtila on aina
päälimmäisenä.

PATENTKRAV

1. Förbättrat förfarande för behandling av tasker i realtid särskilt i fusionerade system, vari taskerna är försedda med prioritet, och av vilka tasker var och en alltid exekveras till slut så, att endast exekveringen av en task med högre prioritet kan göra dess exekvering långsammare, vid vilket förfarande alltid då en transaktion iakttages, dess behandling enligt förfarandet påbörjas genast, fastän föregående transaktions behandling skulle vara oavslutad, ifall detta ej speciellt är förbjudet, varvid en behandlare kan ha flera oavslutade behandlingssekvenser medan den senast påbörjade behandlingssekvensen är i gång, vid vilket förfarande en yttre transaktion förorsakar ett avbrott, varur uppstår en transaktionen beskrivande information och en ny behandlingssekvens genast, då ett avbrott i den igång varande behandlingssekvensen är tillåten, k ä n n e t e c k n a t därav, att genast då avbrottet skett i den bildade behandlingssekvensen:
- 1) förbjuds behandlingen av nya transaktioner, bestäms en första task, som behandlar den ur avbrottet bildade informationen, i den första taskens data antecknas, att den första tasken bör behandla den nya informationen, den första taskens prioritet kontrolleras och, såvida den är högre än den för närvarande gällande prioriteten eller den andra under exekvering varande taskens prioritet, exekveras följande åtgärder 11 - 12;
 - 11) den gällande prioriteten lagras såsom data för den enskilda behandlingen gammal-prioritet; såsom gällande prioritet sätts den första taskens prioritet; behandlingen av nya transaktioner tillåts;
 - 12) så länge den gällande prioriteten är högre än den gamla-prioriteten, förverkligas följande förfaringskede 121;
 - 121) nya transaktioners behandling förbjuds; om det existerar en tredje task, vars prioritet är den gällan-

- de prioriteten och vilken har att behandla informationen, tillåts behandlingen av nya transaktioner och åtgärderna 1211 - 1212 exekveras, annars utbyts den närmast lägre prioriteten till gällande prioritet; och
- 5 behandlingen av nya transaktioner tillåts;
- 1211) den i föregående punkt hittade tredje tasken exekveras; vid exekvringen kan det uppstå ny information, vilka taskerna bör behandla;
- 1212) från var och en bildad information bestäms en
- 10 fjärde denna behandlande task, i den fjärde taskens information antecknas, att den fjärde tasken bör behandla den bildade informationen, den fjärde taskens prioritet kontrolleras och, såvida den är högre än den just gällande prioriteten, ändras den fjärde taskens
- 15 prioritet till gällande prioritet;
- 2) nya transaktioners behandling tillåts och den avbrutna transaktionen fortsätts.

2. Förbättrat förfarande för behandling av tasker i realtid särskilt i fusionerade system, vari
- 20 taskerna är fösedda med behandlingstider, och av vilka tasker var och en alltid exekveras till slut så, att endast en sådan task, som har en tidigare termin, exekvering kan göras långsammare, vid vilket förfarande alltid då en transaktion iakttages, dess behandling
- 25 enligt detta förfarande genast påbörjas, fastän föregående transaktions behandling skulle vara oavslutad, om detta ej speciellt är förbjudet, varvid en behandlare kan ha flera oavslutade behandlingssekvenser medan den senast påbörjade behandlingssekvensen är i gång,
- 30 och vid vilket förfarande en yttre transaktion förorsakar ett avbrott, varur alltid uppstår en transaktionen beskrivande information och en ny behandlingssekvens genast då ett avbrott i den igång varande behandlingssekvensen är tillåten, k ä n n e t e c k n a t därav, att genast då avbrottet skett i den bildade behandlingssekvensen:
- 35

- 1) förbjuds behandlingen av nya transaktioner; bestäms

- en informationen behandlande första task; i den första taskens data antecknas att den första tasken bör behandla den nya informationen; den första taskens behandlingstid och den av klockan angivna närvarande stunden adderas ihop, och det erhållna resultatet lagras såsom transaktionens termin; såvida den nya transaktionens termin är tidigare än den gällande terminen eller den andra under exekvering varande taskens under behandling varande termin, exekveras följande åtgärder
- 11 - 12;
- 11) den gällande terminen lagras såsom data för den enskilda behandlingen gammal-termin; såsom gällande termin sätts den nya transaktionens termin; behandlingen av nya yttre transaktioner tillåts:
- 12) så länge som den gällande terminen är tidigare än den gamla-terminen, upprepas följande förfaringsskede 121;
- 121) nya transaktioners behandling förbjuds; om det existerar en tredje task, vilken har att behandla en information, vars termin är gällande termin, tillåts behandlingen av nya transaktioner och åtgärderna 1211 - 1212 exekveras, annars söks en obehandlad information, vilken i terminsföljden är följande, och dess termin görs till gällande termin och behandlingen av nya transaktioner tillåts,
- 1211) den i föregående punkt hittade tredje tasken exekveras; vid exekveringen kan uppstå ny information, vilka taskerna bör behandla;
- 1212) nutiden avläses från klockan; från var och en bildad information bestäms en transaktionen behandlande fjärde task, i den fjärde taskens information antecknas, att den fjärde tasken bör behandla den bildade informationen, den fjärde taskens behandlingstid kontrolleras och adderas till nutiden och summan antecknas som transaktionens termin och, såvida denna är tidigare än den just gällande terminen, ändras den till gällande termin:

2) nya transaktioners behandling tillåts och den avbrutna transaktionen fortsätts.

3. Förfarande enligt patentkrav 1 eller 2, kännetecknat därav, att arbetsutrymmet för de på exekvering väntande taskerna anordnas i ett stackminne, vari den under exekvering varande taskens arbetsutrymme alltid är överst.

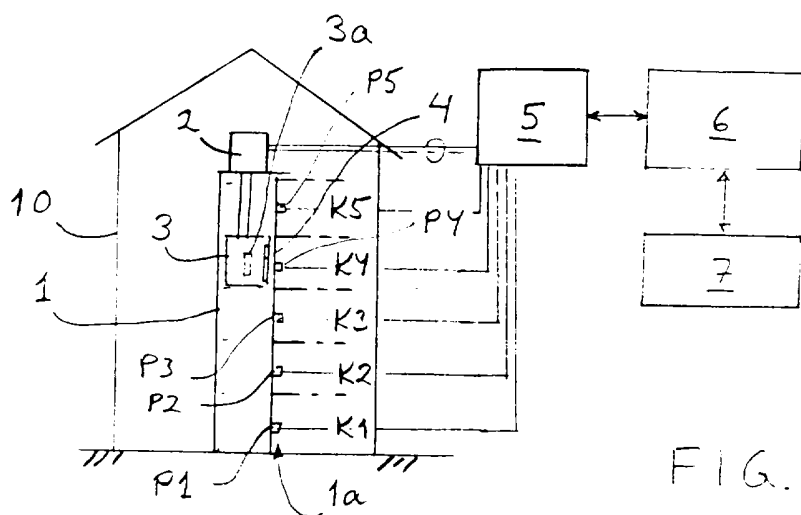
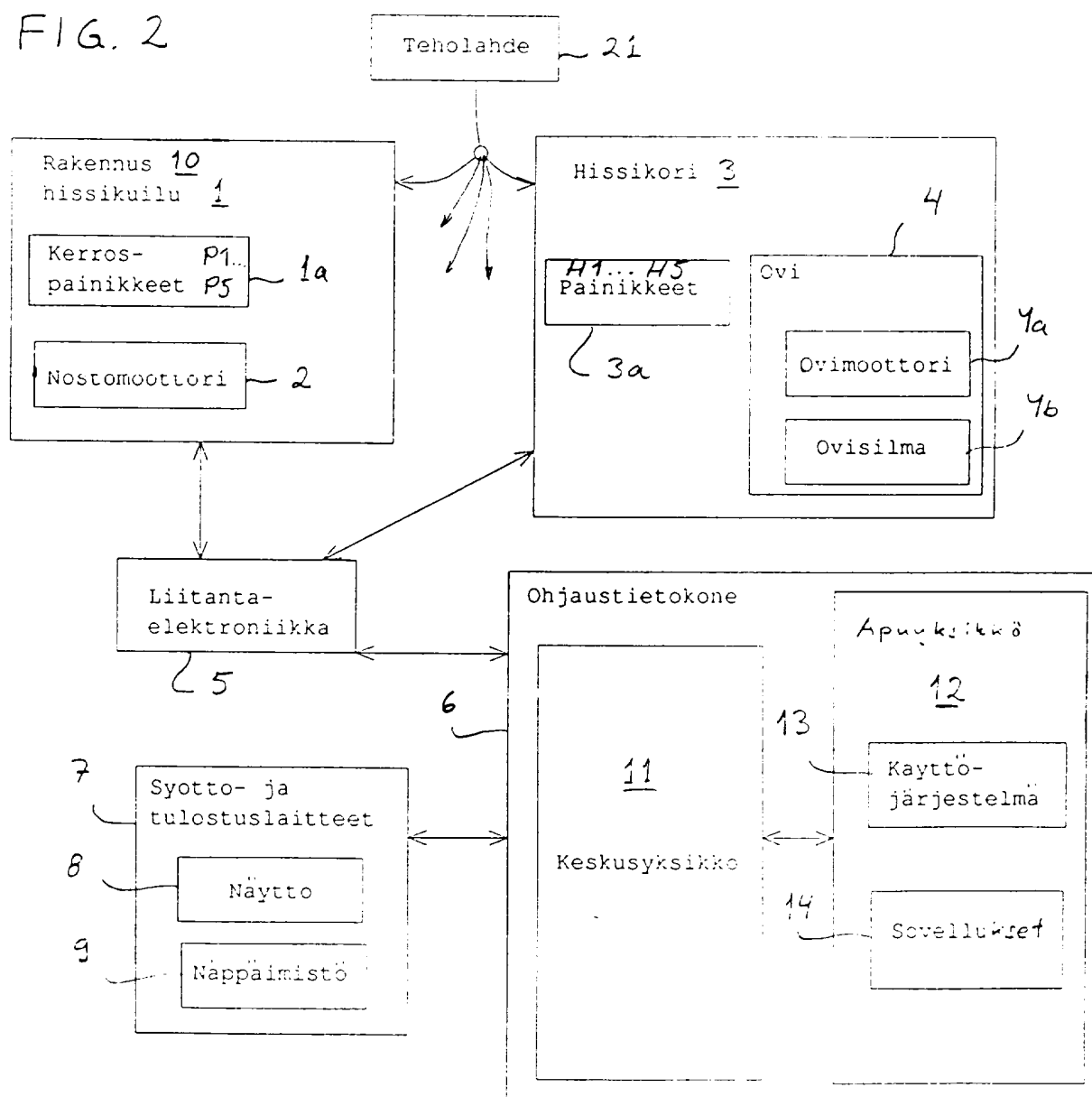


FIG. 2



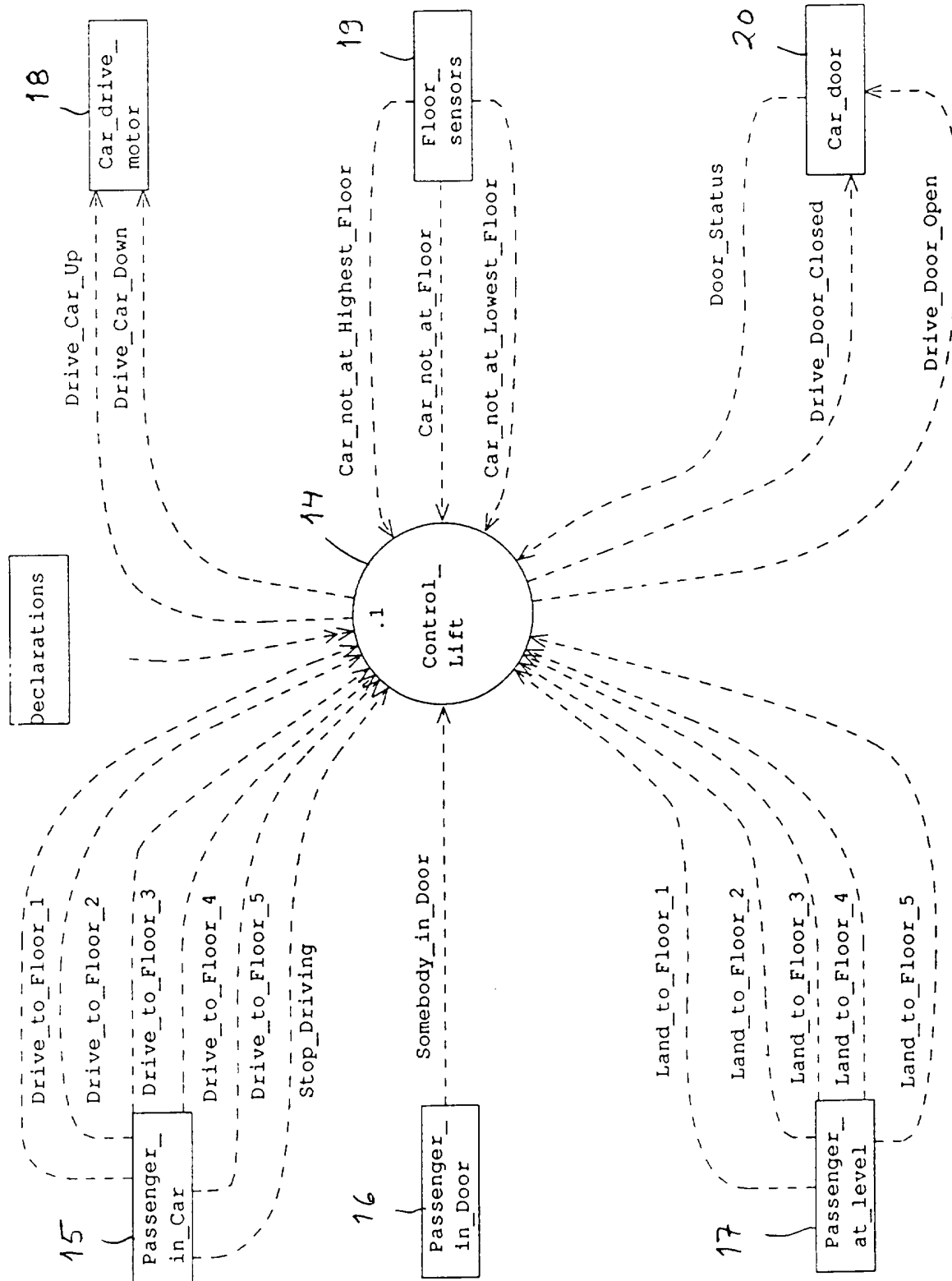


FIG. 3

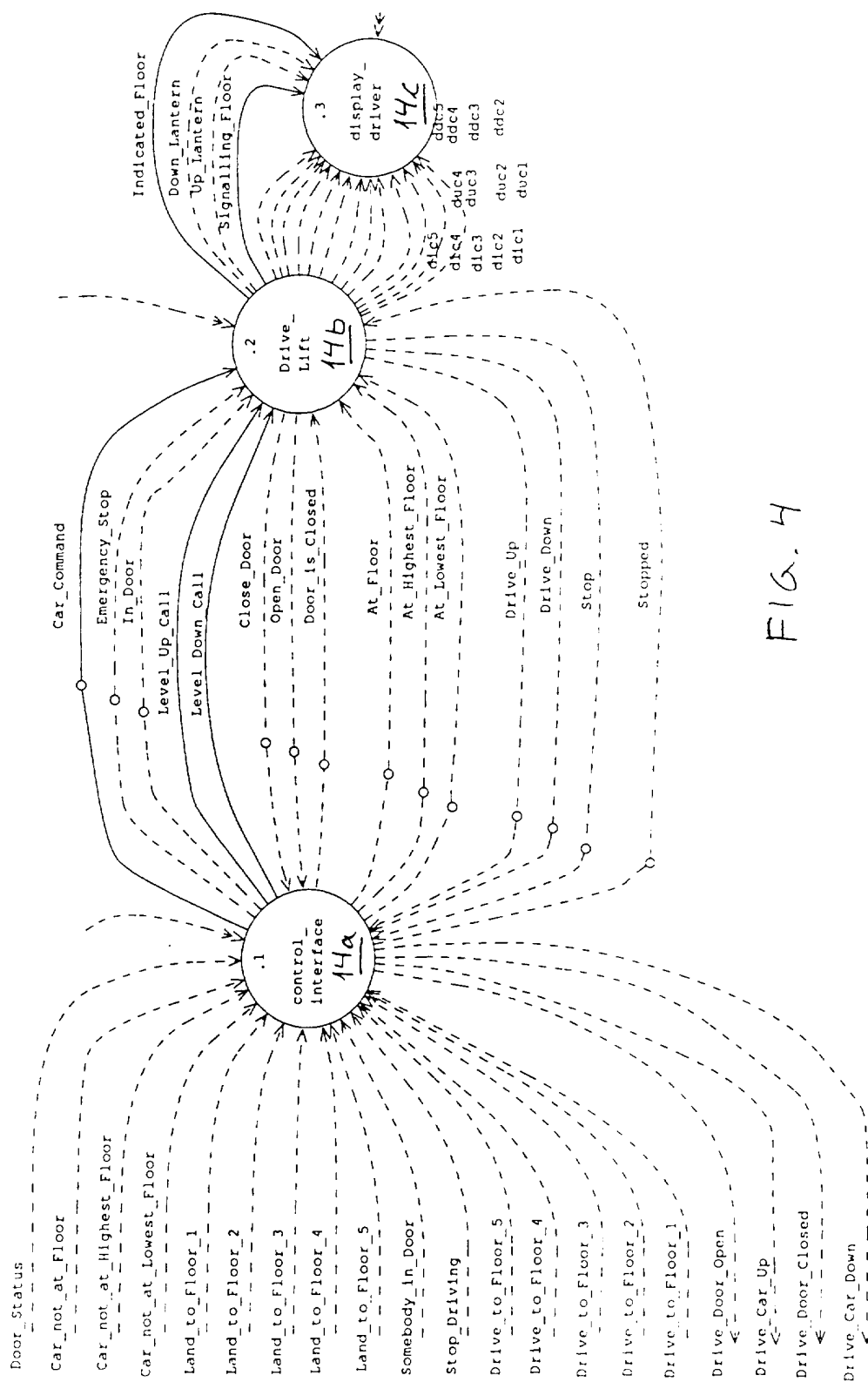


FIG. 4

