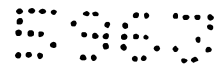


KÖZZÉTÉTELI PÉLDÁNY

P 0200539



Eljárás chipkártya utasításkészletének módosítására

Kivonat

Chipkártya ~~X~~ meglévő utasításkészletének (READ, WRITE) módosítására, illetve kiterjesztésére szolgáló eljárás, amelynek során a bizonyos fájlokkal és a meglévő utasításkészlet parancsaival társított attribútum táblázatokat (~~100, 200, 300~~) használjuk fel az utasítások funkcióinak a módosításához. Ily módon a szabványtól eltérő funkciókat is létre tudunk hozni anélkül, hogy járulékos, pontosabban szabványtól eltérő utasításokat kellene használnunk.

(5. ábra)





Eljárás chipkártya utasításkészletének módosítására

A találmány tárgya eljárás chipkártya utasításkészletének módosítására, mely chipkártya processzort, fájlok tárolására szolgáló memóriát, valamint a fájlokkal társított olyan attribútum (például AC, KEYNR, KEY 1) csoportokat tároló memóriát tartalmaz, ahol mindegyik attribútum az utasításkészlet egy utasításának (például READ) tulajdonságát határozza meg.

A programozható kártyák, amelyeket integrált áramkörös kártyáknak, vagy hétköznapi szóhasználatnál chipkártyáknak is neveznek, fizetések lebonyolításához használatos elektronikus eszközök. A kártya integrált áramköre az adatok feldolgozására mikroprocesszort vagy hasonló elemet foglal magában. A chipkártya processzora általában utasításokat (parancsokat), például READ (OLVASÁS), WRITE (ÍRÁS), UPDATE (FRISSÍTÉS), stb. tartalmazó utasításkészlettel van ellátva, amelyekkel a kártyában lévő (fizetési) adatokkal kapcsolatos műveleteket lehet elvégezni. Ezek az utasításkészletek gyakran szabványos készletek, mint például az EN726 jelzésű európai szabványban meghatározott készlet, ám az ilyen utasításkészletek szükségképpen behatároltak az utasítások számát és célját tekintve. A chipkártyákat is magában foglaló elektronikus fizetési rendszerek fejlődésével együtt igen gyakran lép fel az igény a chipkártyák új funkciókkal való ellátására, vagy a meglévő funkciók módosítására. Ezek a változtatások általában a chipkártya utasításkészletének a módosítását jelentik, ami általános esetben a kártya integrált áramkörének újratervezését is jelenti, ami tulajdonképpen nem nagyon költséges dolog, de a meglévő chipkártyákat használhatatlanná teszi. Ezen túlmenően egy szabványos utasításkészlet módosítása sértené a kártya kompatibilitását a meglévő szabványokkal.

A fentiek miatt igen jelentős igény áll fenn a chipkártyákban lévő utasításkészletek módosítása iránt, anélkül, hogy a létező utasításkészletektől lényegesen el kellene térni.

A találmánnyal célunk az ismert megoldások fenti hiányosságainak a kiküszöbölése, és olyan eljárás létrehozása, amellyel lehetségessé válik egy chipkártya utasításkészletének a módosítása további parancsok hozzáadása nélkül. Ezen kívül célunk olyan új funkciók biztosítása egy chipkártya számára, mely új funkciók az eredeti utasításkészleten alapulnak.

A kitűzött feladatot ennek megfelelően egy chipkártya utasításkészletének módosítására alkalmas eljárással oldottuk meg, mely chipkártya processzort, fájlok tárolására szolgáló memóriát, valamint a fájlokkal társított olyan attribútum (például AC, KEYNR, KEY 1) csoportokat tároló memóriát tartalmaz, ahol mindegyik attribútum az utasításkészlet egy utasításának (például READ) tulajdonságát határozza meg, és lényege, hogy az attribútum csoportba a módosítandó utasításnak megfelelő és az illető utasítás módosított tulajdonságát meghatározó új attribútumot (például NEW FUNCTION) illesztünk.

A fentiekből kitűnően a találmány lényege az, hogy az utasításokat attribútumaik, jellemzőik megváltoztatásával módosítjuk. Mivel az attribútumokat meghatározott fájlokkal társított csoportokba szervezhetjük, a találmány szerinti eljárás lehetővé teszi, hogy az utasításokat szelektív módon, azaz meghatározott fájlokra nézve módosítsuk. Egy attribútum jellemzően egy hozzáférési feltételt specifikál vagy a vonatkozó utasítás által használt meghatározott kulcsot ad meg. Természetesen egy attribútum gépi kódot (vagy egy gépi kódra irányított mutatót) is tartalmazhat, amely egy további műveletet ad meg, és ez a művelet hajt végre egy meghatározott funkciót (például a szükséges titkosítást). Ha egy attribútumba egy új műveletet, vagy egy módosított műveletet illesztünk, az utasítás funkcióját magának az utasításnak a módosítása nélkül változtathatjuk meg.

Természetesen egy attribútum beillesztése egy már meglévő attribútum cseréjét is magában foglalhatja. Általánosságban elmondható, hogy az attribútumok tárolása céljából memóriahelyek tartalékolhatók. Egy új attribútum beillesztése lényegében annyit jelent, hogy ezt az új attribútumot a megfelelő memóriahelyen eltároljuk, és ebben az esetben természetesen minden, az említett memóriahelyen korábban tárolt információt felülírunk.

A találmány szerinti eljárás olyan helyzetekben is alkalmazható, ahol az utasításkészlet attribútum csoportjai úgynevezett első attribútumokat tartalmazó első táblázatot, és úgynevezett második attribútumokat tartalmazó második táblázatot foglalnak magukban. Az első táblázat a második táblázathoz tartozó referenciákat tartalmaz, illetve azt mondja meg, hogy a második táblázatban hová illesztettünk új attribútumot. Egy ilyen első táblázat például a meghatározott kulcsot tartalmazhatja, míg a második táblázat a vonatkozó utasítással összefüggésben használandó műveleteket tartalmazza. Megjegyzendő, hogy az első táblázatra referenciákat tartalmazó alaptáblázattal is utalhatunk. Így, ha a második táblázatba egy új attribútumot illesztünk, a módosítást az attribútumok minimális változásával hajthattuk végre.

A korszerű chipkártyák legalább két különböző adatfájlt tartalmazhatnak, például a chipkártya különböző alkalmazásához tartozó adatfájlokat. Ilyen esetben egy attribútum csoport a vonatkozó adatfájlokkal társított alcsoportokat tartalmazhat. A találmány lehetővé teszi, hogy az új attribútumot egy korlátozott számú alcsoportba illesszük be, és ezzel az utasítás módosításokat az egyes adatfájlokra specifikusan hajthassuk végre. Ezen a módon biztosítani tudjuk az utasítások nagy rugalmasságát.

A találmányt az alábbiakban a csatolt rajz segítségével ismertetjük részletesebben, amelyen a javasolt eljárás példakénti fogatosítási módjának jellemző részleteit tüntettük fel. A rajzon az

1. ábra fizetési és azonosítási célokra használható chipkártya nézete, a
2. ábrán például az 1. ábra szerinti chipkártya integrált áramkörének vázlatos tömbvázlatát tüntettük fel, a
3. ábrán egy chipkártya, például a 2. ábra szerinti felépítésű chipkártya utasítás-készletének attribútum táblázata látható, a
4. ábra a 3. ábrán bemutatott attribútum táblázattal társított kulcs táblázat vázlat, az
5. ábra a 3. ábra szerinti attribútum táblázattal és a 4. ábra szerinti kulcs táblázattal társított eljárástáblázat vázlata, és a
6. ábrán egy chipkártya hierarchikus fájlstruktúrájának lehetséges kialakítása látható.

Az 1. ábrán vázlatosan, példaként bemutatott 1 chipkártya 2 hordozóelemébe integrált áramkör van beágyazva, amely kártyaolvasó berendezéshez vagy hasonló készülékhez 3 érintkezőkkel rendelkezik. Külön felhívjuk a figyelmet arra, hogy találmányunk a bemutatott példától eltérően ún. érintkezésmentes chipkártyákhoz is használható.

Az 1. ábrán csupán vázlatosan feltüntetett integrált áramkört a 2. ábrán 10 hivatkozási számmal jelöltük, és az ábra szerint 11 processzort, 12 memóriát és 13 bemeneti/kimeneti áramkört tartalmaz. A 12 memória az adatokat átmenetileg tároló RAM részt, valamint adatok állandó vagy közel állandó tárolására alkalmas ROM részt tartalmazhat. Ez utóbbi műszakilag célszerűen EEPROM-ként valósítható meg. A ROM részben tárolt adatok mind programadatokat (utasításokat, programokat), mind pedig fizetési adatokat, vagyis a

pénzügyi tranzakciókra vonatkozó adatokat tartalmazhat. Természetesen a 11 processzor utasításkészletének az eltárolására az ábrán nem látható külön memóriát is biztosíthatunk.

A 11 processzor szabványos, például az EN726 jelzésű európai szabványnak megfelelő utasításkészletet tartalmaz, de ez nem kötelező, a 11 processzor más, szabványtól eltérő utasításkészlettel is használható. A bemutatott példában használt utasításkészletet a 3. ábra segítségével mutatjuk be részletesebben.

A 13 bemeneti/kimeneti áramkör bonyolítja le a 11 processzor és az ábrán nem látható külső eszköz (eszközök) közötti kommunikációt. Ilyen eszköz lehet egy szabványos kártyaolvasó berendezés is. A 13 bemeneti/kimeneti áramkör, amely megfelelő pufferfokozatokat is tartalmazhat, adatbuszon keresztül kapcsolódik a 11 processzorhoz. A 11 processzor és a 12 memória ugyanazon adatbuszon keresztül kommunikál egymással.

A 3. ábrán vázlatosan és csupán példaként bemutatott 100 attribútum táblázat különböző részeket tartalmaz, amely különböző fájlokhoz társulnak. Az 1 chipkártya egy lehetséges fájlstruktúráját a későbbiekben a 6. ábra kapcsán külön is bemutatjuk. A 100 attribútum táblázat egy első, A fájlal (a 6. ábrán 21 fájl) társított része 111, 121 és 131 mezőt tartalmaz. A 111 mező egy fájljelzőt tartalmaz, amely az ábrán "FILE A" feliratként látható. A 121 mező egy utasítástáblázatot tartalmaz, például az 1 chipkártya utasításkészletét tartalmazó táblázatot, vagy legalább az utasításkészlet utasításaira vonatkozó referenciákat. Nyilvánvaló, hogy az utasítások megfelelően tartalmazzák a 11 processzor által végrehajtott gépi kódokat. A bemutatott példában három ilyen utasítást, nevezetesen "READ", "WRITE" és "UPDATE" utasítást tüntettünk fel. A harmadik 131 mező tartalmazza a 121 mező utasításainak az attribútumait. Ez azt jelenti, hogy a második 121 mező minden egyes utasításához tartalmazhat egy bejegyzést a harmadik 131 mező, és ez a bejegyzés olyan attribútum csoportot jelent, amely pedig az utasítás funkcióját határozza meg. Az attribútumok például hozzáférési feltételeket és kulcsszámokat tartalmazhatnak, melyeket a 3. ábrán "AC" és "KEYNR" jelölésként tüntettünk fel. A hozzáférési feltételek például olyanokat és hasonlókat tartalmazhatnak, mint "ALWAYS" (mindig), "NEVER" (soha), "PROTECTED" (védett). A kulcsszámok a 4. ábrán bemutatott kulcstáblázatban lévő bejegyzésekre hivatkozhatnak. A 100 attribútum táblázat harmadik 131 mezője így olyan attribútum alaptáblázatot képezhet, amely (a 4. ábrán látható) további táblázatokra hivatkozik.

Jóllehet a 3. ábrán bemutatott 100 attribútum táblázat első, második és harmadik 111, 121 és 131 mezői a jobb érthetőség kedvéért egymással szomszédos mezőkként vannak feltüntetve, tényleges kialakításoknál ezek a 111, 121, 131 mezők megfelelően tárolhatók az 1 chipkártya 12 memóriájának különböző, egymással nem szomszédos részein is.

A 131 mezőben felsorolt attribútumok a 111 mezőben meghatározott módon a "FILE A" fájlhoz tartoznak, ami azt jelenti, hogy a 131 mező attribútumai kizárólag erre a fájlra érvényesek. Ez a szerkezet lehetővé teszi, hogy az utasítások különböző (adat) fájlok tekintetében különböző funkciókat jelentsenek. Egy ilyen fájl például adatokat tartalmazhat az 1 chipkártya meghatározott alkalmazására (funkciójára) vonatkozóan. Jellemző alkalmazások a "pénztárca" fizetések lebonyolítására vagy pénzüsszegek felvételére, a "hűség program" vásárlói vagy fogyasztói hűségpontok gyűjtésére, vagy "hozzáférés", azaz épületekbe való belépés, és hasonló. Egyetlen ilyen alkalmazáson belül is lehetőség van egynél több adatfájl használatára, ezért egynél több társított attribútum csoport létezhet. Az egy adatfájlra jellemző attribútumoknak köszönhetően egy utasítás funkciója, legalábbis az attribútumok által meghatározott részekre, egy adatfájlra vonatkozhat. Így az "READ" utasítás hozzáférési feltételei eltérők lehetnek például egy pénztárca adatokat tartalmazó fájl, és az 1 chipkártya tulajdonosának személyi adatait tartalmazó fájl esetében. Ezt a 4. és 5. ábra segítségével magyarázzuk meg.

Az aktuális hozzáférési feltételek, amelyeket a 3. ábrán jelképesen "AC" módon jelöltünk, vagy az alapvető 100 attribútum táblázatban, vagy pedig az ábrákon nem látható külön táblázatban vannak eltárolva. Hasonlóképpen a 4. ábrán látható kulcsok (KEY 1, KEY 2, ...) ugyancsak a 100 attribútum táblázatban lehetnek. A találmány lényegének ismertetéséhez abból indulunk ki, hogy a "KEYNR" bejegyzések, amelyek egyszerű számok (1, 2, ...) lehetnek, a 4. ábra 200 kulcstáblázatára hivatkoznak.

A 4. ábra 200 kulcstáblázatának minden egyes bejegyzése tartalmaz egy kulcsszámot (KEYNR), egy kulcsot (KEY 1, KEY 2, ...), és egy műveleti azonosítót vagy műveleti számot (a 4. és 5. ábrán "PRONR"-ként jelölve). Minden egyes műveleti azonosító a 300 művelet táblázat egy bejegyzésére vonatkozik. A 300 művelet táblázat minden egyes bejegyzése tartalmaz egy műveleti számot és egy műveletet, amelyet az 5. ábrán például "DES", vagy "RSA" módon jelöltünk be. Egy ilyen művelet lehet egy kódoló művelet vagy valamilyen más adاتمűvelet.

A találmánynak megfelelően az 1 chipkártya utasításait az attribútumok változtatásával módosíthatjuk. Így tehát ha a 300 művelet táblázatban módosítunk egy bejegyzést, akkor egy új funkciót kapunk, vagy egy már meglévő funkciót módosítunk. Az 5. ábrán a 300 művelet táblázat második bejegyzésére egy új funkciót viszünk be, például "szorozd meg az adatot egy közös tényezővel", vagy "az adat értékéből vonj ki 1-et", stb. Így tehát, ha a 200 kulcstáblázatban kiválasztjuk a 3-as számú kulcsot, akkor a műveleti szám az új funkcióként ott lévő 2-es számú műveletre fog mutatni. Minden alkalommal, amikor olyan utasítást hajtunk végre, amely használja a KEY 3 kulcsot, már az új funkciót hajtjuk végre. Természetesen egy új funkcióra történő utalásokat különböző módon valósíthatjuk meg, például közvetlenül a 3. ábrán bemutatott 100 attribútum táblázatból. Hasonlóképpen a 200 kulcstáblázat "álkulcsokat" is tartalmazhat, hogy lehetővé tegye a 200 kulcstáblázat alkalmazását anélkül, hogy éppen egy kulcsot használnánk.

Ahelyett, hogy az új funkciót a 300 művelet táblázatba írjuk be, egy mutatót is elhelyezhetünk, amely arra a memóriahelyre mutat, ahol az új vagy módosított kódot eltároltuk.

Mint az előzőekben kifejtettük, a bázisként szolgáló attribútum táblázatban, tehát a 3. ábrán bemutatott 100 attribútum táblázatban a különböző fájlokhoz különböző attribútumok létezhetnek. A találmány szerinti eljárás az egyes adatfájlokhoz tartozó attribútumok módosítása révén igen rugalmas utasításkészletet tesz lehetővé.

A chipkártyák a bennük tárolt fájlok szervezésére hierarchikus könyvtár struktúrát tartalmazhatnak. Ilyen fájlkönyvtár struktúrát tüntettünk fel a 6. ábrán. 20 fő fájlnak három 21, 22, 23 alfájlla van, ami azt jelenti, hogy a 20 fő fájl referenciákat tartalmazhat a 21, 22, 23 alfájlokhoz, vagy az ilyen referenciák valahol máshol (például egy közös könyvtárban) vannak eltárolva. A 20 fő fájlal attribútum csoport van társítva. A 3. ábrán bemutatott módon lehetőség van arra, hogy mindhárom 21, 22, 23 alfájllhoz egyéni attribútum csoportok tartozzanak, azonban ha egy attribútum csoport a 20 fő fájlhoz van társítva, akkor ez az attribútum csoport a hierarchiában az összes az alatti, tehát a 20 fő fájl alatti 21, 22, 23, stb. alfájl részére hozzáférhető és alkalmazható. Más szavakkal ez azt jelenti, hogy ugyanazt az attribútum csoportot társíthatjuk a 21, 22, 23 alfájlokhoz. Ha tehát egy utasítást megváltoztatunk oly módon, hogy egy meghatározott fájlal társított attribútumot (vagy attribútum csoportot) módosítunk, ez a módosítás érvényes lehet az említett fájl alatt a hierarchiában lejjebb lévő összes fájlra. Ezáltal lehetővé válik, hogy egy utasítást olyan meghatározott

chipkártya alkalmazáshoz meg tudjunk változtatni, mely alkalmazás több fájl is magában foglal.

Mint kifejtettük, az utasítások módosítását a találmány értelmében úgy érjük el, hogy a meglévő attribútumokat helyettesítjük és/vagy módosítjuk. A javasolt eljárás így lehetővé teszi, hogy szabványos utasításokat (szabványos kártya funkciókat) használjunk fel szabványtól eltérő műveletek végrehajtására vagy befolyásolására. A találmány a fentiek értelmében egy chipkártya módosított utasításkészletét, valamint magát az ilyen módosított utasításkészlettel ellátott chipkártyát is érinti és tartalmazza. Egy új attribútum attribútum táblázatba való behelyezését az új attribútumnak egy megfelelő memóriahelyen történő eltárolása révén hajtjuk végre. Az új attribútumot képviselő adatot egy számítógépről kártyaolvasó, illetve kártyaíró segítségével vihetjük át a chipkártya memóriájába. Visszautalva a 2. ábrára, az adatokat a 13 bemeneti/kimeneti áramkör segítségével hagyományos módon írjuk a 12 memóriába.

Szakember számára nyilvánvaló, hogy az eddig bemutatott megvalósítási módok és kiviteli alakok kizárólag a találmány lényegének megértését elősegítő példák, és az igénypontok által meghatározott oltalmi körön belül számos, a bemutatottól eltérő kialakítás is létrehozható.

Szabadalmi igénypontok

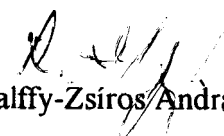
1. Eljárás chipkártya utasításkészletének módosítására, mely chipkártya (1) processzort (11), fájlok (21, 22) tárolására szolgáló memóriát (12), valamint a fájlokkal (21, 22) társított olyan attribútum (például AC, KEYNR, KEY 1) csoportokat tároló memóriát (12) tartalmaz, ahol mindegyik attribútum az utasításkészlet egy utasításának (például READ) tulajdonságát határozza meg, azzal jellemezve, hogy az attribútum csoportba a módosítandó utasításnak megfelelő és az illető utasítás módosított tulajdonságát meghatározó új attribútumot (például NEW FUNCTION) illesztünk.
2. Az 1. igénypont szerinti eljárás, azzal jellemezve, hogy az attribútum csoportokat egy további táblázatra (200, 300) hivatkozó első táblázatba (100) szervezzük, és az új attribútumot (NEW FUNCTION) egy további táblázatba (300) illesztjük.

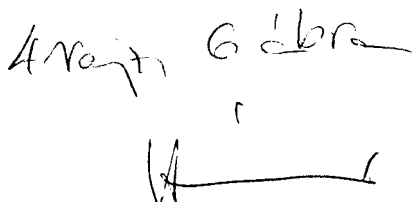
3. Az 1. vagy 2. igénypont szerinti eljárás, azzal jellemezve, hogy az új attribútum beillesztése során egy meglévő attribútumot helyettesítünk.
4. Az 1. - 3. igénypontok bármelyike szerinti eljárás, azzal jellemezve, hogy a chipkártya (1) memóriájában (12) legalább két fájlt (21, 22), továbbá az egyes fájlokkal (21, 22) társított attribútum csoportot és korlátozott számú csoportba illesztett új attribútumot tárolunk.
5. Az 1. - 4. igénypontok bármelyike szerinti eljárás, azzal jellemezve, hogy a fájlokat (21, 22) hierarchikus módon szervezzük, ahol minden egyes attribútum csoportot legalább egy fájlal (20) társítunk és minden további fájlt (21, 22) hierarchikusan az említett fájl (20) alá osztunk be.
6. Az 1. - 5. igénypontok bármelyike szerinti eljárás, azzal jellemezve, hogy a chipkártya (1) minden egyes fájljával társítunk egy utasításkészletet.
7. Utasításkészlet chipkártyához (1), azzal jellemezve, hogy az 1. - 6. igénypontok bármelyike szerinti eljárással módosított utasításkészletként van kialakítva.
8. Chipkártya, amely memóriát és utasításkészlettel rendelkező processzort tartalmaz, azzal jellemezve, hogy az 1. - 6. igénypontok bármelyike szerinti eljárással módosított utasításkészletet tartalmaz.

A meghatalmazott:

DANUBIA

Szabadalmi és Védjegy Iroda Kft.


Antalffy-Zsíros András



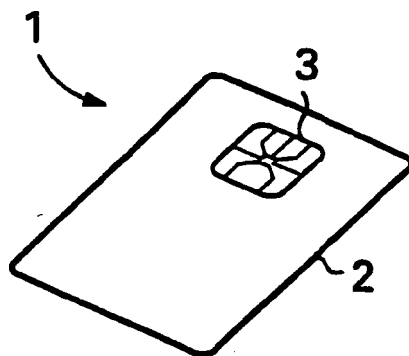


Fig. 1

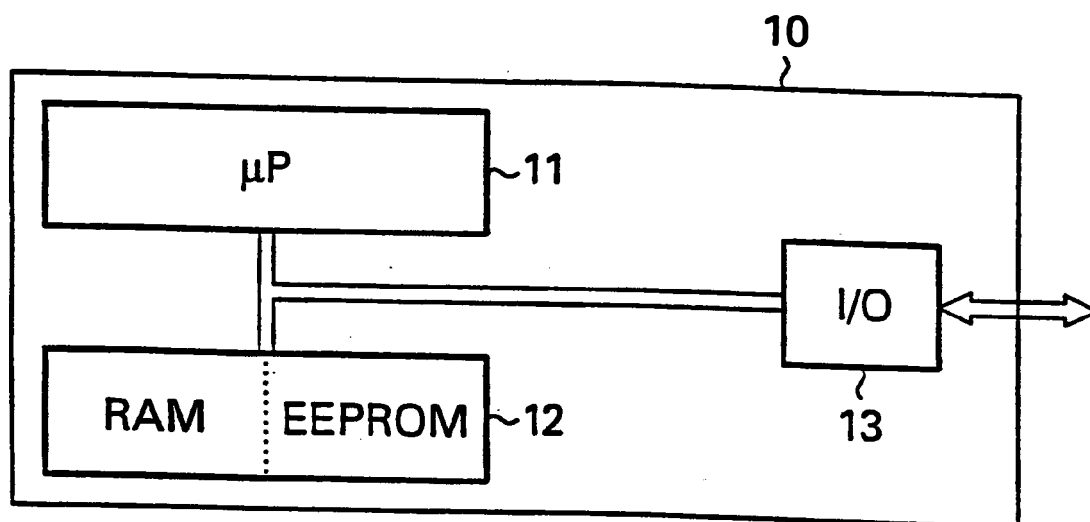


Fig. 2

KÖZZÉTÉTELI
PÉLDÁNY

2/4

100
?

ATTRIBUTE TABLE		
File A	READ WRITE UPDATE . . .	[AC], [KEYNR.] [AC], [KEYNR.] [AC], [KEYNR.] . . .
<u>111</u>	<u>121</u>	<u>131</u>
File B	READ WRITE	[AC], [KEYNR.] [AC], [KEYNR.]
<u>112</u>	<u>122</u>	<u>132</u>
....		

Fig. 3

KÖZZÉTÉTELI
PÉLDÁNY

3/4

200

KEY TABLE		
KEYNR.	KEY	PRONR.
1	KEY 1	1
2	KEY 2	3
3	KEY 3	2
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.

Fig. 4

300

PROCEDURE TABLE	
PRONR.	PROCEDURE
1	[DES]
2	[NEW FUNCTION]
3	[RSA]
.	.
.	.
.	.
.	.
.	.

Fig. 5

KÖZZÉTÉTELI 4/4
PÉLDÁNY

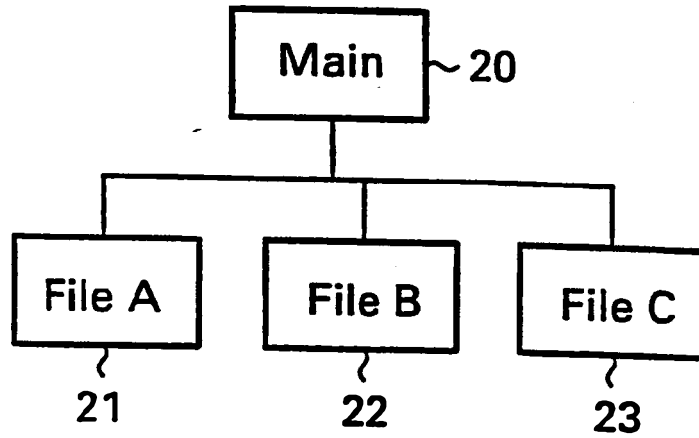


Fig. 6