



MINISTERO DELLO SVILUPPO ECONOMICO  
DIREZIONE GENERALE PER LA TUTELA DELLA PROPRIETÀ INDUSTRIALE  
UFFICIO ITALIANO BREVETTI E MARCHI

# UTBM

|                           |                        |
|---------------------------|------------------------|
| <b>DOMANDA NUMERO</b>     | <b>101982900000603</b> |
| <b>Data Deposito</b>      | <b>22/11/1982</b>      |
| <b>Data Pubblicazione</b> | <b>22/05/1984</b>      |

|                               |                  |
|-------------------------------|------------------|
| <b>Priorità</b>               | <b>323812</b>    |
| <b>Nazione Priorità</b>       | <b>US</b>        |
| <b>Data Deposito Priorità</b> | <b>23-NOV-81</b> |

Titolo

PROCEDIMENTO E APPARECCHIATURA PER INTRODURRE VARIAZIONI DI PROGRAMMA IN SISTEMI CONTROLLATI A PROGRAMMA

# DOCUMENTAZIONE RILEGATA

"PROCEDIMENTO E APPARECCHIATURA PER INTRODURRE VARIAZIONI DI PROGRAMMA  
IN SISTEMI CONTROLLATI A PROGRAMMA".

WESTERN ELECTRIC COMPANY, INCORPORATED, a  
New York, N.Y. (U.S.A.).



Inventori Designati: Thomas Patrick BISHOP e Susan Jean PICUS.

\* \* \* \* \*

Depositata il

22 NOV. 1982

al No. 24362 A/82

=.=.=.=.=

RIASSUNTO

Disposizione per impiegare o un programma originale (132) oppure una versione modificata (142) del programma in un sistema controllato da programma. Ogniqualvolta una versione modificata (142) di una funzione di programma selezionata viene memorizzata nel sistema, essa è etichettata con un numero sequenziale di funzione (143) e la funzione di programma originale è ritenuta nel sistema. Ogniqualvolta viene avviato un processo di programma, esso è etichettato con un numero di sequenza di processo (111). I due tipi di numeri di sequenza sono assegnati da una sorgente comune (100). Ogniqualvolta la funzione di programma selezionata è richiesta da un processo di programma, viene eseguita una comparazione (144) fra il numero di sequenza di processo (111) e il numero di sequenza di funzione (143). Se la comparazione indica che il processo è stato iniziato prima della memorizzazione nel sistema della modifica, allora viene eseguita la versione originale della funzione di programma selezionata; in caso contrario viene eseguita la versione modificata del programma selezionato. Perciò, la versione originale oppure la versione modificata

risulta coerentemente disponibile durante tutta la durata attiva di un processo.

=.=.=.=.

#### DESCRIZIONE

La presente invenzione riguarda un procedimento per introdurre variazioni in un sistema controllato dal programma memorizzato, senza richiedere arresto parziale o totale del sistema e successiva inizializzazione.

I sistemi controllati da programma sono stati applicati ad un'ampia gamma di applicazioni di elaborazione e controllo di dati. La flessibilità funzionale di questi sistemi deriva dal programma, memorizzato nella memoria di questi sistemi, che controlla il loro funzionamento.

In sistemi o impianti controllati da programma, è frequentemente necessario introdurre variazioni di programma dopo l'installazione iniziale per l'adattamento di nuovi servizi e/o nuovo hardware. Il modo più comune per introdurre questi cambiamenti è quello di arrestare il sistema e inizializzarlo completamente. Sfortunatamente, ciò rende il sistema incapace di fornire servizio per l'intervallo di inizializzazione, solitamente un periodo di vari minuti o anche ore. In aggiunta, l'inizializzazione distrugge frequentemente alcune porzioni di dati di più lungo termine che possono influenzare le prestazioni successive del sistema; ad esempio, essa può azzerare dati forniti da un utente telefonico definenti un certo aspetto del proprio servizio. L'inizializzazione è particolarmente indesiderabile se il sistema controllato da programma controlla processi continui, come ad esempio processi chimici; piani accurati devono essere ese-

guiti per introdurre tale inizializzazione in corrispondenza di un momento in cui i processi chimici controllati sono stati tutti arrestati o hanno raggiunto un certo equilibrio.

Un metodo meno drastico per introdurre cambiamenti di programma con inizializzazione parziale è attuato, ad esempio, in impianti di commutazione telefonica, come ad esempio l'Impianto di Commutazione Elettronico No. 1 descritto nel Bell System Technical Journal, Settembre, Edizione 1964 e nel brevetto statunitense N. 3.570.008. Le caratteristiche speciali di questi sistemi hanno reso possibile introdurre variazioni senza disturbare quegli utenti che sono già collegati ad altri utenti. Tutti i dati relativi a connessioni completate e a servizi di utente sono isolati e trattati in modo speciale tramite programmi di inizializzazione. Questi sistemi hanno processori duplicati, uno dei quali è del tipo "in linea" e l'altro dei quali è un'unità di riserva pronta per assumere il controllo. Una versione modificata di un programma viene caricata in uno dei processori, mentre il processore si trova nel modo a riposo. Esso viene quindi commutato "in linea" e esegue un programma di inizializzazione che non modifica nessuno dei dati associati con connessioni completate o servizi di utente. La versione modificata del programma è quindi pure caricata nell'altro processore, ed il sistema si trova in uno stato stabile. Utenti che stavano eseguendo una connessione non avranno le loro chiamate completate, e perciò questo tipo di cambiamento viene attuato durante ore di attività molto ridotta dell'impianto o sistema per minimizzare i disturbi. Questo metodo di introdurre variazioni non è applicabile ai sistemi controllati da programma moderni più sofisticati. Questi sistemi



sono normalmente fatti funzionare sotto il controllo di un programma complesso chiamato "sistema operativo" che avvia altri programmi chiamati "processi". L'esecuzione effettiva dei compiti del sistema o dell'impianto è normalmente attuata tramite i processi tramite l'esecuzione di una serie di segmenti di programma chiamati "funzioni" di programma. L'esecuzione delle funzioni di programma può essere avviata mediante una chiamata di funzione da un qualsiasi processo nel sistema. Le funzioni di programma accedono ad una base di dati comune, grandi porzioni della quale sono controllate da molti processi, in cui i concatenamenti fra i vari elementi di dati associati sono molteplici e complessi. Queste strutture di dati complesse rendono drasticamente complicato il tipo di trattamento speciale di una porzione selezionata di dati impiegata nei sistemi della tecnica nota summenzionati. In tal modo, l'inizializzazione di taluni processi e nel contempo la ritenzione di altri, mantenendo al tempo stesso una base di dati coerente, è estremamente difficoltosa. E' inoltre impossibile prevedere la natura delle variazioni del programma e pianificare queste variazioni nella progettazione della base di dati. Per tutte le ragioni precedenti, si è rivelato difficile introdurre tutte le variazioni di programma, tranne quelle secondarie, nei sistemi o impianti controllati da programma moderni più sofisticati senza inizializzazione massiccia.

Questo problema è stato risolto dalla presente invenzione, il cui metodo comprende le fasi di generare un numero sequenziale di sistema che è variato solo in modo monotono in una data direzione, assegnare il numero sequenziale di sistema ad un processo con un numero sequenziale di processo in corrispondenza del momento in cui il processo viene avviato,

memorizzare una versione modificata di una funzione di programma selezionata nella memoria del programma, ritenendo al tempo stesso una versione attuale della funzione di programma selezionata, registrare un numero di sequenza di funzione decisionale, il numero di sequenza di funzione decisionale essendo oltre il numero sequenziale o di sequenza del sistema nella data direzione o dato senso, variare il numero sequenziale del sistema nella data direzione in modo tale che esso abbia a raggiungere un valore progredito almeno nella data direzione come il numero sequenziale di funzione decisionale, e eseguire la versione attuale se il numero di sequenza o sequenziale di processo è prima del numero sequenziale di funzione decisionale nella data direzione, e eseguire la versione modificata se il numero sequenziale di processo è almeno progredito come il numero sequenziale di funzione decisionale nella data direzione.

Uno scopo della presente invenzione è quello di consentire la introduzione di variazioni di programma in sistemi controllati da programma senza interferire con i processi in corso di esecuzione, anche mentre il sistema sta funzionando con carico normale.

Secondo la presente invenzione, cambiamenti sono introdotti nel programma di un sistema o impianto controllato da programma memorizzando una versione modificata di una funzione di programma, ritenendo nel contempo la versione attuale e facendo quindi sì che tutti i processi avviati prima della variazione abbiano ad eseguire la versione attuale di tale funzione, e facendo in modo che tutti i processi avviati dopo la variazione abbiano ad eseguire la versione modificata di detta funzione.

Secondo una forma di realizzazione, ciò viene attuato con

l'ausilio di un numero sequenziale di sistema, mantenuto dal sistema; Un numero sequenziale di sistema viene variato solo in modo monotono in una direzione o dato senso, e viene variato almeno ogni volta che una funzione di programma modificata viene introdotta nel sistema. Il valore attuale del numero sequenziale del sistema viene registrato con un processo ogniqualvolta il processo viene avviato. Ogniqualvolta una versione modificata di una funzione di programma viene memorizzata nel sistema, un numero sequenziale di funzione decisionale viene assegnato alla funzione modificata, avente un valore il quale è oltre nella data direzione, rispetto al valore del numero sequenziale di processo di un qualsiasi processo attualmente attivo che impiega tale funzione di programma. Il numero sequenziale di sistema viene successivamente modificato, in modo tale che esso abbia a trovarsi nella data direzione almeno progredito come il numero sequenziale di funzione decisionale. Successivamente, ogniqualvolta un processo richiede la funzione che è stata modificata, il numero sequenziale di funzione decisionale viene comparato con il numero di processo. Se il risultato della comparazione indica che la versione modificata è stata memorizzata nel sistema prima dell'inizio del processo di chiamata, allora viene eseguita la versione modificata della funzione di programma; in caso contrario viene eseguita la funzione attuale.

In una forma di realizzazione specifica dell'invenzione, funzioni di programma possono essere richieste da un processo tramite l'impiego di un vettore in una tabella vettoriale di trasferimento memorizzata, che normalmente punta direttamente alla funzione di programma desiderata. Se viene introdotta una funzione di programma modificata nel sistema, al-



lora il vettore, invece punterà ad una funzione decisionale. La funzione decisionale esegue quindi la decisione relativa al fatto se deve essere eseguita la versione attuale oppure la versione modificata della funzione di programma. Se deve essere eseguita la versione attuale, allora viene eseguita una diramazione di programma alla versione attuale; in caso contrario viene eseguita una diramazione di programma alla versione modificata. Vantaggiosamente, la funzione decisionale può precedere direttamente la versione modificata, semplificando così il processo di recupero di spazio di memorizzazione quando versioni vecchie di funzione di programma non sono più necessarie.

Periodicamente, un programma di audit può essere eseguito per determinare se un qualsiasi processo attivo riguardante una funzione di programma, che è stata modificata, ha un numero sequenziale che è, nella data direzione, prima di quello della versione modificata della funzione di programma. Se non vi sono processi di questo tipo, allora la versione attuale della funzione di programma può essere cancellata, il vettore può puntare direttamente alla versione modificata della funzione invece di puntare alla funzione decisionale, e il blocco di memoria occupato dalla versione attuale della funzione di programma diviene spazio disponibile per altre funzioni di programma nuove o modificate.

Vantaggiosamente, i numeri sequenziali possono essere costituiti da qualsiasi numero a gruppo di numeri che variano in modo monotono, o tramite incrementazione, oppure tramite decrementazione, per cui è possibile indicare se un dato numero sequenziale era assegnato prima di un altro numero sequenziale. Numeri sequenziali devono variare con una fre-

quenza almeno pari a quella con cui sono introdotte le variazioni di programma e possono variare molto più frequentemente. Ad esempio, è possibile impiegare una quantità rappresentante data e ora come un numero sequenziale, poichè questa quantità si mantiene variabile in modo monotono tramite incrementazione positiva. Ciò presenta l'ulteriore vantaggio di impiegare un numero sequenziale che può avere impiego aggiuntivo, ad esempio per rivelare processi che sono stati in stato attivo per un periodo di tempo eccessivo. Inoltre, se è desiderato caricare una modifica un certo tempo prima che la modifica debba essere attivata, allora il numero sequenziale di funzione decisionale può essere preimpostato ad una quantità rappresentante il tempo di attivazione; la modifica non sarà in tal caso impiegata prima del citato tempo di attivazione.

L'invenzione sarà meglio compresa dalla seguente descrizione particolareggiata, quando letta in unione ai disegni in cui:

la figura 1 è una disposizione di memoria di una forma di realizzazione illustrativa dell'invenzione, rappresentante le relazioni di indirizzamento fra versioni attuale e modificata di una funzione di programma;

la figura 2 è uno schema a blocchi di un processore di un sistema o impianto controllato da programma; e

la figura 3 è un diagramma di flusso di un programma di audit per recuperare memorizzazione di una versione vecchia di un programma quando non più richiesta.

Un sistema o impianto controllato da programma ha due tipi di informazione memorizzata nella sua memoria e cioè: programmi e dati. Un

programma comprende un insieme di istruzioni basilari di un processore, ciascuna delle quali istruzioni dirige una certa fase elementare, per cui un insieme quando eseguito realizza un qualche compito più ampio del sistema o serie di compiti. I dati comprendono informazione relativa allo stato attuale di un compito, informazione ottenuta da sorgenti esterne e preliminarmente o precedentemente memorizzata nel sistema, e informazione generata per eseguire i compiti funzionali del sistema. I programmi eseguono i loro compiti elaborando dati e controllando sistemi di ingresso/uscita in conformità con i valori attuali dei dati.

In sistemi controllati da programma moderni, il controllo basilare del sistema risiede in un programma principale il quale, assieme ai dati associati, viene chiamato il sistema operativo. Il sistema operativo dirige l'esecuzione dei compiti del sistema creando e attuando processi.

Un processo è un programma più memoria di dati di controllo associata, chiamato blocco di controllo di processo. Il blocco di controllo di processo mantiene traccia o registrazione del contesto di ciascun processo includente informazione come ad esempio l'identificazione del processo, lo stato corrente del processo, la priorità di processo, il tempo del sistema in corrispondenza dell'inizio del processo, ecc. I processi, a loro volta, eseguono i loro compiti richiedendo l'esecuzione di una serie di funzioni di programma.

Funzionando normalmente, un sistema controllato da programma avrà solitamente alcuni processi che sono attivi e alcuni che sono a riposo. Un processo è considerato come attivo durante il periodo in cui esso sta attivamente eseguendo un compito, attendendo un blocco o periodo di tempo

per la continuazione della propria esecuzione di un compito, o attendendo dispositivi di ingresso oppure di uscita. In seguito al completamento di un compito, o gruppo di compiti, esso assume lo stato di riposo e rimane in questo stato finchè esso non viene nuovamente avviato per eseguire un altro compito o gruppo di compiti. Quando un processo entra in uno stato di riposo, un minimo di dati viene riportato per l'impiego successivo al riavviamento del processo. Nessun riferimento di programma diretto è ritenuto quando un processo passa a riposo.

In corrispondenza di uno stadio qualsiasi nella vita di un processo, una variazione di programma può essere introdotta memorizzando una funzione di programma modificata in un certo spazio disponibile in memoria. Tuttavia, risultati non corretti possono essere generati se una funzione di programma modificata viene impiegata da un processo che era già attivo prima della introduzione della variazione. Al fine di garantire che ciascun processo abbia ad impiegare la medesima funzione di programma durante tutto il tempo in cui esso rimane attivo, funzioni decisionali sono associate con ciascuna versione modificata di una funzione di programma. Le funzioni decisionali sono brevi programmi che determinano quando il processo che ha richiesto la funzione di programma è stato iniziato - prima o dopo del caricamento nel sistema o impianto della versione modificata della funzione di programma. Se la funzione modificata è stata caricata prima dell'avviamento del processo, allora viene impiegata la versione modificata stessa; se la versione modificata è stata caricata dopo l'avviamento del processo, allora viene impiegata la versione attuale.

Un numero sequenziale del sistema è mantenuto per tener trac-



cia e registrazione dell'ordine di eventi, cioè se un processo è stato iniziato prima o dopo il caricamento nel sistema di una versione modificata di una funzione di programma. Un numero sequenziale di sistema può essere costituito da una qualsiasi quantità variata in modo monotono, in modo tale che può essere stabilito l'ordinamento nel tempo tra due numeri sequenziali diversi. Ad esempio, se un numero sequenziale di sistema è sempre incrementato in modo positivo, allora un numero sequenziale di sistema maggiore è associato con un istante temporale maggiormente posteriore che non un numero sequenziale di sistema inferiore. Il tempo, comprendente data e tempo del giorno viene impiegato come un numero sequenziale di sistema in questa forma di realizzazione.

Nella figura 1, il blocco 100 contiene il numero sequenziale del sistema, il tempo, aggiornato automaticamente dal sistema operativo impiegando l'orologio del sistema. L'orologio del sistema costituisce parte del processore 200. Esso è ben noto nella tecnica e non è rappresentato nel disegno. Il meccanismo per aggiornare una entrata temporale in memoria è pure noto, e non sarà ulteriormente descritto nel presente contesto.

Un numero sequenziale di processo rappresentante tempo viene memorizzato nel blocco di controllo di processo di un processo ogniqualvolta esso viene avviato. Il blocco 110 di controllo di processo contiene l'informazione di controllo di processo associata con un processo tipico. Come si è detto precedentemente, il blocco di controllo di processo include vari elementi di informazione associati col processo. La posizione 111 del blocco 110, contiene il numero sequenziale di processo



che indica il momento in corrispondenza del quale è stato avviato il processo associato. Ogniqualvolta un processo diviene attivo, il valore attuale del numero sequenziale del sistema viene posto nella posizione 111. Un numero sequenziale di funzione decisionale, rappresentante il tempo attuale incrementato di un secondo, viene assegnato alla funzione decisionale in corrispondenza del momento in cui una funzione di programma modificata viene caricata nella memoria.

Ciascuno dei blocchi funzionali 130, 140 e 150 di figura 1 rappresenta un'area di memoria per memorizzare un programma. Ciascuno ha un'area per memorizzare una funzione decisionale (131, 141 e 151) e un'area per memorizzare una funzione di programma (132, 142 e 152). Le funzioni di programma rappresentano una versione attuale,  $F$ , (132), una versione modificata,  $F'$ , (142) ed una seconda modifica,  $F''$  (152). Il blocco 141 di funzione decisionale comprende un programma 144 di funzione decisionale e un numero sequenziale 143 di funzione decisionale identificato come  $m$ . Il programma di funzione decisionale può essere rappresentato come segue: se il numero sequenziale di processo,  $n(pc)$ , è inferiore al numero sequenziale di funzione decisionale,  $m$ , allora il programma passa a  $f(2)$  (blocco 132); in caso contrario va a  $f'(2)$  (blocco 142). In altre parole, se il processo è stato avviato prima della introduzione della variazione, allora viene impiegata la versione attuale della funzione di programma,  $F$ ; in caso contrario impiega la versione modificata della funzione di programma,  $F'$ .

I programmi sono assegnati a qualsiasi segmento disponibile di memoria. Funzioni di programma individuali possono essere impiegate da

molti processi, ma memorizzate solo una volta per risparmiare spazio di memoria utile. Poichè molti programmi possono richiamare una funzione di programma, è mantenuta una lista degli indirizzi attuali di tutte le funzioni di programma ed essa è resa disponibile per tutti i programmi chiamanti. Questa lista viene chiamata tabella vettoriale di trasferimento. Quando una variazione viene caricata nel sistema, l'indirizzo della tabella vettoriale di trasferimento viene aggiornato per puntare alla funzione decisionale associata con la versione modificata. Quando un processo richiede la funzione di programma che è stata modificata, la funzione decisionale puntata dalla tabella vettoriale di trasferimento determinerà se deve essere impiegata la versione attuale o modificata della funzione di programma. Nella tabella vettoriale di trasferimento 120, l'entrata 121 punta all'indirizzo della funzione di programma o funzione decisionale esemplificative. Se solo una versione di un programma viene caricata nel sistema, allora l'entrata 121 rappresenta l'indirizzo della funzione di programma. Se più di una versione viene caricata nel sistema, allora l'entrata 121 rappresenta l'indirizzo di una funzione decisionale che consentirà la selezione della versione corretta del programma. Quando la versione modificata della funzione di programma  $F'$  viene caricata nella memoria del sistema, l'entrata della tabella vettoriale in corrispondenza di 121 viene variata da  $f(2)$  che è l'indirizzo della versione attuale della funzione di programma, a  $f'(1)$ , che è l'indirizzo della funzione decisionale associata con  $F'$ . Come risultato, prima che sia eseguita l'una o l'altra versione della funzione di programma, la funzione decisionale compara i numeri sequenziali di processo e funzione decisionale per decidere se il pro-

cesso è stato iniziato prima o dopo l'introduzione della variazione. Se il processo è stato avviato prima della variazione, allora si verifica una diramazione alla versione attuale  $F$ ; in caso contrario viene eseguita la versione modificata  $F'$ .

Se sussiste una qualche necessità di avere una modifica addizionale, allora deve essere introdotta una seconda modifica rappresentata da  $F''$  nel blocco 150. Quando  $F''$  viene caricata nel sistema, l'entrata 121 nella tabella vettoriale di trasferimento viene variata per puntare a  $f''(1)$ , che è la funzione decisionale associata con  $F''$ . Il blocco 150 comprende la funzione decisionale, blocco 151, e la funzione di programma,  $F''$ , blocco 152. Il blocco della funzione decisionale comprende il numero sequenziale 153 di funzione decisionale, rappresentato da  $p$ , ed il programma 154 di funzione decisionale. Il programma 154 di funzione decisionale può essere dichiarato come segue: se il numero sequenziale di processo  $n(pc)$ , è inferiore al numero sequenziale di funzione decisionale,  $p$ , allora va a  $f'(1)$ ; in caso contrario va a  $f''(2)$ . In altre parole, se il processo è stato avviato prima del caricamento di  $F''$ , allora viene eseguito un salto al programma 144 di funzione decisionale associato con la versione  $F'$ ; in caso contrario viene eseguita  $F''$ . Il risultato del programma 144 di funzione decisionale è un salto alla funzione originale,  $F$ , se il numero sequenziale di processo,  $n(pc)$  è inferiore al numero sequenziale di funzione decisionale  $m$ ; in caso contrario viene eseguita la funzione di programma  $F'$ . In questo esempio, i processi impiegano le funzioni di programma  $F$ ,  $F'$ , o  $F''$ . I processi più vecchi impiegano  $F$ , i processi intermedi impiegano  $F'$  e i processi avviati più recentemente impiegano  $F''$ . Se è necessario o



desiderato modificare simultaneamente un gruppo di funzioni, allora una funzione decisionale viene inserita per ciascuna di queste funzioni di programma, e il medesimo numero sequenziale viene assegnato a tutte le funzioni decisionali associate con tale gruppo. E' pure possibile attivazione ritardata di una modifica di programma. Il numero sequenziale di funzione decisionale associato con tale modifica sarebbe il tempo di attivazione ritardato. La modifica non avrebbe quindi effetto sino al tempo di attivazione.

Questa descrizione è stata fornita nei termini di una modifica di una funzione di programma singola. Manifestamente, molte funzioni di programma possono essere modificate impiegando le tecniche della presente invenzione. Modificazioni di programma per molte funzioni di programma diverse, caricate in corrispondenza di momenti diversi, ciascuna funzione di programma modificata avendo la sua propria funzione decisionale associata, possono esistere contemporaneamente nel sistema.

La figura 2 illustra tre componenti principali di un sistema controllato da programma, e cioè il processore centrale 202, la memoria principale 201 e l'apparecchiatura di ingresso/uscita 203. L'unità di elaborazione centrale e la sua memoria associata possono essere uno qualsiasi di una pluralità di moderni processori ben noti nella tecnica. L'apparecchiatura di ingresso/uscita può includere dispositivi di memoria supplementare come ad esempio unità a nastro magnetico e dischi a testine fisse oppure mobili. L'unità di elaborazione centrale esegue processi e funzioni di programma che sono memorizzati in memoria. Nell'eseguire questi processi e funzioni di programma, l'unità di elaborazione centrale accede e mo-

difica dati memorizzati nella memoria. L'unità di elaborazione centrale esegue semplici passi come ad esempio la comparazione di due numeri, l'esecuzione di un ramo condizionato basato su tale comparazione, l'incrementazione di una quantità nella memoria, lo spostamento di quantità memorizzate in memoria da una posizione all'altra, la diramazione incondizionata ad un indirizzo indicato direttamente oppure indirettamente. L'unità di elaborazione centrale è pure in grado di caricare una versione modificata di una funzione di programma dalla apparecchiatura di ingresso/uscita 203 nella memoria principale 201.

Periodicamente, vengono eseguiti programmi di audit sia per recuperare spazio di memoria occupato da versioni vecchie delle funzioni di programma che non sono più impiegate sia per aggiornare la tabella vettoriale di trasferimento con indirizzi di versioni modificate di funzioni di programma, al fine di bypassare la funzione decisionale non più richiesta. La figura 3 è un diagramma di flusso di un programma di audit che esamina processi che potrebbero impiegare o la funzione di programma vecchia F oppure la versione modificata F'. In questo esempio si suppone che il programma di audit sia eseguito dopo l'introduzione della modifica F' nel sistema e prima che abbia a verificarsi la modifica F''.

Il programma di audit può essere considerato come suddiviso in tre parti e cioè: passi iniziali (300-302), ciclo (303-306) e recupero memoria (307-308). Il programma di audit viene avviato tramite il sistema operativo. Il programma di audit deve leggere il numero sequenziale di funzione decisionale che deve essere comparato col numero sequenziale di processo di tutti i processi attivi che potrebbero impiegare la versione at-



tuale o modificata della funzione di programma. Viene quindi fatto accesso al primo processo. La serie di passi del ciclo inizia con un controllo per determinare se il processo è in stato di riposo o inattivo. Se il processo non è in stato inattivo, allora il numero sequenziale di processo viene comparato col numero sequenziale di funzione decisionale. Se il numero sequenziale di processo non è almeno pari al numero sequenziale di funzione decisionale, allora il programma di audit viene terminato; ciò significa che un qualche processo attivo sta ancora impiegando la versione attuale della funzione di programma. In caso contrario, il processo ciclico continua finchè tutti i processi appropriati non sono stati esaminati. Se il programma di audit non è stato terminato dopo che tutti i processi appropriati sono stati esaminati, allora il programma di audit può procedere per recuperare memoria. Esso fa ciò variando l'entrata della tabella vettoriale in modo che punto direttamente alla versione modificata, e rendendo disponibile lo spazio di memoria occupato dalla versione corrente per il sistema operativo per la riassegnazione per altri scopi.

Come è illustrato nel blocco 300 del diagramma di flusso di figura 3, il programma di audit inizia l'esecuzione dopo essere stato avviato dal sistema operativo. Il numero sequenziale di funzione decisionale,  $m$ , associato con  $F'$ , viene letto dal programma di audit in 301. In 302, il programma di audit accede al primo processo che impiega  $F$  o  $F'$ . Quindi, in corrispondenza dei punti decisionali 303 e 304, rispettivamente, il programma di audit determina se il processo è inattivo e se il suo numero sequenziale,  $n(pc)$ , è maggiore del o uguale al numero sequenziale di funzione decisionale associato con  $F'$ . Se entrambe queste decisioni sono negative,



allora il programma di audit conclude che il processo è attivo e impiega F, la versione attuale della funzione di programma. Poichè F è ancora impiegata da un processo attivo, essa non può essere cancellata. Conseguentemente, il programma di audit termina in corrispondenza di 309.

Se l'una o l'altra decisione è positiva, allora il programma di audit conclude che il processo è o a riposo o inattivo e preparato per impiegare la versione modificata della funzione di programma F' quando essa viene avviata, oppure sta già impiegando F'. Il programma di audit procede al punto decisionale 305 ove esso determina se vi sono ulteriori processi che impiegano F o F'. Se vi sono altri di questi processi, allora il programma di audit accede al processo successivo nel blocco 306, e cicla attraverso i punti decisionali 303, 304 e 305 nuovamente.

Il programma di audit ripete il ciclo finchè tutti i processi che impiegano le funzioni F o F' non sono stati esaminati. Se non sono stati scoperti processi attivi impieganti F, allora non vi è ragione di conservare l'indirizzo della funzione decisionale nella tabella vettoriale di trasferimento. Perciò, nel blocco 307, il programma di audit cambia l'entrata 121 nella tabella di trasferimento per puntare direttamente all'indirizzo f'(2) della nuova funzione di programma F'.

Poichè la funzione di programma F non è più puntata da alcun processo e non può più essere indirizzata dalla funzione decisionale, essa occupa spazio sprecato in memoria. Poichè questo spazio può essere impiegato per un qualche altro scopo, il programma di audit cancella la funzione di programma F, rilascia il blocco 130 come uno spazio di memoria libero in corrispondenza di 308 e termina in corrispondenza di 309. Si deve no-

tare che il blocco 130 include pure lo spazio precedentemente occupato da una funzione decisionale quando F è stata introdotta come una modifica del programma. Ritendendo o assicura la modifica e la sua funzione decisionale associata, l'interblocco di memoria può essere reso disponibile quando F non è più richiesta.

In taluni sistemi può essere difficile che il programma di audit possa determinare se un processo è attivo o inattivo. In questo caso, il controllo del blocco 303 può essere bypassato. Può pure essere più conveniente esaminare tutti i processi indipendentemente dal fatto se essi impieghino o no F o F'. Gli unici risultati nocivi di tale semplificazione sono che la funzione decisionale rimane attiva per un periodo maggiore di quello necessario, e che lo spazio di memorizzazione occupata dalla versione attuale della funzione di programma non può essere altrettanto rapidamente recuperato.

Una grande classe di modifiche di programma possono essere introdotte impiegando le tecniche della presente invenzione, fra cui modifiche in programmi che avviano e eseguono processi. In linea generale, questi programmi sono funzioni di programma richieste dal sistema operativo che è pure un gruppo di funzioni di programma. Qualsiasi programma richiesto indirettamente può in linea di principio, essere variato impiegando le tecniche della presente invenzione. Tuttavia, talune restrizioni sorgono nell'eseguire modificazioni a funzioni di programma che impiegano dati interpretati differentemente dalle versioni attuale e modificata di una funzione di programma, in particolare quando questi dati sono conservati durante un intervallo inattivo di un processo oppure quando questi dati sono

condivisi da due o più processi. Anche quando si accede a questi dati, un impiego accurato di funzioni decisionali per dati, congiuntamente a funzioni decisionali per funzioni di programma, può consentire l'introduzione di modifiche in questi dati impiegando le tecniche della presente invenzione. Si deve notare che un prerequisito implicito è un unico punto di accesso come ad esempio un vettore di trasferimento per funzione di programma, impiegato ogniqualvolta una funzione decisionale viene impiegata per accedere a questi dati.

Si deve tener presente che la forma di realizzazione precedentemente descritta è semplicemente illustrativa dei principi della presente invenzione; altre disposizioni possono peraltro essere concepite dagli esperti del ramo senza allontanarsi dallo spirito e ambito protettivo dell'invenzione.

#### R I V E N D I C A Z I O N I

1. Procedimento per introdurre variazioni in un sistema o impianto controllato da programma senza richiedere arresto parziale o totale del sistema e successiva inizializzazione, comprendente le fasi di

generare un numero sequenziale di sistema il quale viene variato solo in modo monotono in una data direzione o in un dato senso;

assegnare il numero sequenziale di sistema ad un processo come un numero sequenziale di processo in corrispondenza del momento in cui il processo viene avviato;

memorizzare una versione modificata di una funzione di programma selezionata nella memoria di programma, conservando al tempo stesso una versione attuale della funzione di programma selezionata;

registrare un numero sequenziale di funzione decisionale, il numero sequenziale di funzione decisionale essendo oltre il numero sequenziale di sistema o dell'impianto nella data direzione;

variare il numero sequenziale del sistema nella data direzione in maniera tale che esso abbia a raggiungere un valore almeno progredito nella data direzione come il numero sequenziale di funzione decisionale; e

eseguire la versione attuale se il numero sequenziale di processo precede il numero sequenziale di funzione decisionale nella data direzione, e eseguire la versione modificata se il numero sequenziale di processo è almeno progredito come il numero sequenziale di funzione decisionale nella data direzione.

2. Procedimento secondo la rivendicazione 1, in cui la fase di esecuzione comprende inoltre le fasi di:

comparare il numero sequenziale di processo col numero sequenziale di funzione decisionale;

diramare alla versione attuale se il risultato della comparazione indica che il numero sequenziale di processo si trova prima del numero di funzione decisionale nella data direzione; e

diramare alla versione modificata se il risultato della comparazione indica che il numero sequenziale di processo si trova progredito almeno come il numero sequenziale di funzione decisionale nella data direzione.

3. Procedimento secondo la rivendicazione 1, in cui la fase di assegnazione comprende la fase di:

memorizzare il numero sequenziale del sistema o dell'impianto



in una posizione di memoria associata con il processo.

4. Procedimento secondo la rivendicazione 1, in cui detta versione attuale di detta funzione di programma selezionata comprende una versione intermedia ed una versione più vecchia di detta funzione di programma selezionato, e in cui detta fase di diramazione a detta versione attuale, se risultato di detta comparazione, indica che detto numero sequenziale di processo è prima di detto numero sequenziale di funzione decisionale in detta data direzione, comprende inoltre le fasi di:

eseguire una seconda comparazione fra detto numero sequenziale di processo e un secondo numero sequenziale di funzione decisionale associato con detta versione intermedia di detta funzione di programma selezionata;

diramare a detta versione più vecchia se detta seconda comparazione indica che detto numero sequenziale di processo si trova prima di detto secondo numero sequenziale di funzione decisionale in detta data direzione; e

diramare a detta versione intermedia se detta seconda comparazione indica che detto numero sequenziale di processo è almeno progredito come detto secondo numero sequenziale di funzione decisionale in detta data direzione.

5. Procedimento secondo le rivendicazioni 1, 2, 3 oppure 4 comprendente inoltre le fasi di:

ripetere detta fase di assegnazione per una pluralità di processi, compresi tutti i processi che realizzano chiamate funzionali a detta funzione di programma selezionata;

eseguire periodicamente terze comparazioni fra il numero sequenziale di processo di almeno tutti i processi attivi inclusi in detta pluralità di processi e detto numero sequenziale di funzione decisionale; e cancellare detta versione attuale se nessuno di detti numeri sequenziali di processo di almeno tutti i processi attivi inclusi in detta pluralità di processi si trova prima di detto numero sequenziale di funzione decisionale in detta data direzione.

6. Sistema a processore controllato da programma comprendente:

un'unità di elaborazione centrale (202) ed una memoria di programma (201) caratterizzato dal fatto che la memoria di programma (201) contiene una versione attuale di una funzione di programma selezionata (132); e memorizza una versione modificata della funzione di programma selezionata (142);

l'unità di elaborazione centrale (202) interroga la memoria di programma (201, 110, 111, 121, 141, 132, 142) e esegue la versione attuale di una funzione di programma selezionata se un processo richiedente la funzione di programma selezionata è stato avviato prima della memorizzazione della versione modificata nella memoria, e esegue la versione modificata se il processo richiedente la funzione di programma selezionata è stato iniziato dopo la memorizzazione in memoria della versione modificata.

7. Sistema o impianto a processore controllato da programma secondo la rivendicazione 6, caratterizzato dal fatto che:

è memorizzato un numero sequenziale del sistema, il quale numero viene variato in un dato senso o data direzione;

il numero sequenziale del sistema è memorizzato come un nume-

ro sequenziale di processo in corrispondenza dell'istante di inizio del processo;

un numero sequenziale di funzione decisionale è memorizzato in corrispondenza del momento della memorizzazione della versione modificata, il numero sequenziale di funzione decisionale essendo oltre il numero sequenziale di processo nella data direzione;

il numero sequenziale di sistema essendo modificato in modo tale che esso abbia a raggiungere un valore almeno progredito nella data direzione come il numero sequenziale di funzione decisionale;

l'unità di elaborazione centrale (202) esegue la versione modificata della funzione selezionata in risposta ad una chiamata di funzione ad una funzione selezionata se il numero sequenziale di processo è prima del numero sequenziale di funzione decisionale nella data direzione ed esegue la versione attuale della funzione selezionata in risposta ad una chiamata di funzione alla funzione selezionata se il numero sequenziale di processo è almeno progredito come il numero funzionale o di funzione decisionale nella data direzione.

8. Sistema o impianto a processore controllato da programma secondo la rivendicazione 7, caratterizzato dal fatto che:

l'unità di elaborazione centrale (202) è atta a comparare il numero sequenziale di processo con il numero sequenziale di funzione decisionale e a diramare alla versione attuale se la comparazione indica che il numero sequenziale di processo è prima del numero sequenziale di funzione decisionale nella data direzione e a diramare alla versione modificata se la comparazione indica che il numero sequenziale di processo

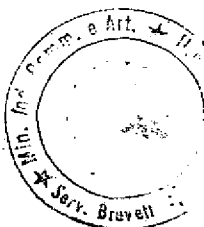
è almeno progredito come il numero sequenziale di funzione decisionale nella data direzione.

9. Sistema o impianto a processore controllato da programma secondo la rivendicazione 8, caratterizzato dal fatto che la memoria (201) del programma memorizza una versione intermedia della funzione di programma selezionata ed una versione più vecchia al posto della versione attuale;

l'unità di elaborazione centrale (202) dirama la versione attuale se la comparazione indica che il numero sequenziale di processo si trova prima del numero sequenziale di funzione decisionale nella data direzione ed esegue una seconda comparazione fra il numero sequenziale di processo e il secondo numero sequenziale di funzione decisionale associato con la versione intermedia della funzione di programma selezionata e dirama alla versione più vecchia se la seconda comparazione indica che il numero sequenziale di processo si trova prima del secondo numero sequenziale di funzione decisionale nella data direzione e dirama alla versione intermedia se la seconda comparazione indica che il numero sequenziale di processo si trova almeno progredito come il secondo numero di funzione decisionale nella data direzione.

Il Mandatario:

  
- Dr. Ing. G. MODIANO -



Ufficiale Rogante  
Pietro Messineo

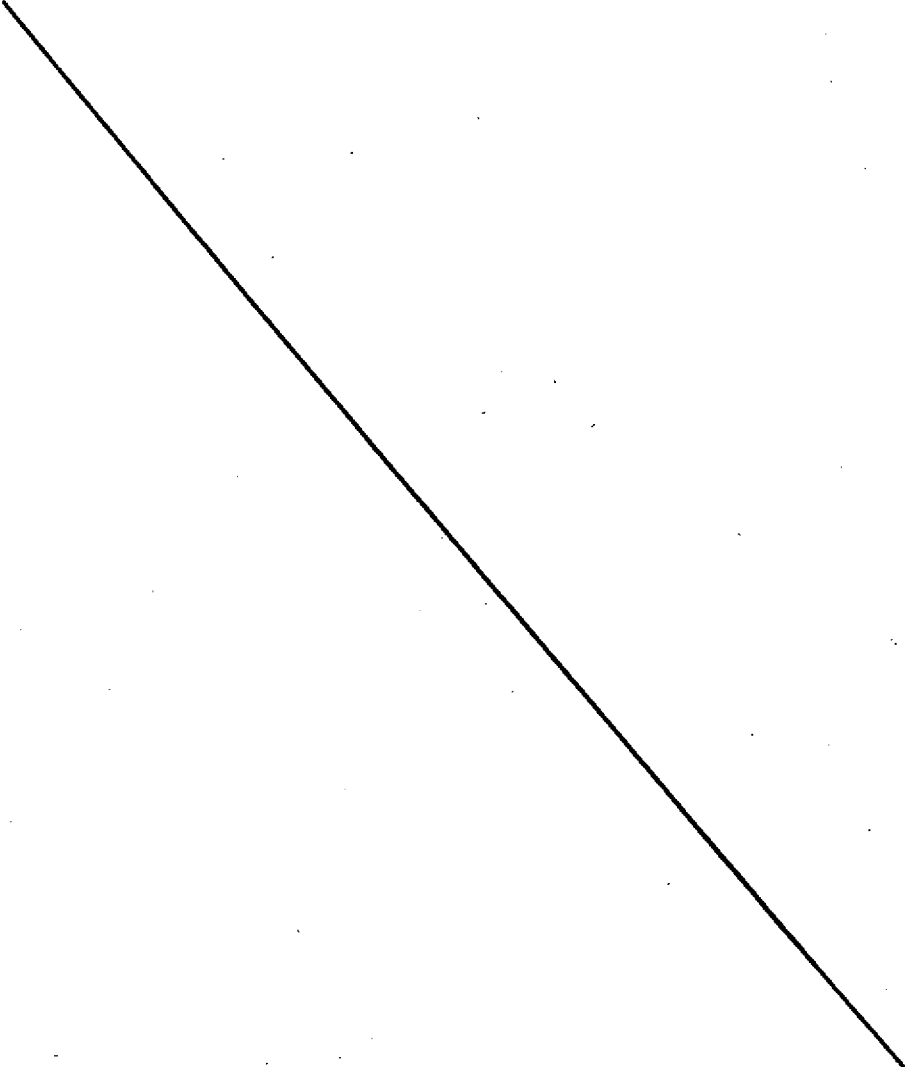
23 Novembre 1981

323812

PROCEDIMENTO ED APPARECCHIATURA PER INTRODURRE VARIAZIONI  
DI PROGRAMMA IN IMPIANTI CONTROLLATI A PROGRAMMA

Campo tecnico

Questa invenzione riguarda dei procedimenti per l'introduzione di variazioni di programma in impianti di programma controllato. Più particolarmente riguarda uno schema per introdurre tali variazioni in un impianto durante l'esecuzione di programma senza richiedere l'inizializzazione dell'impianto.



risulta coerentemente disponibile durante tutta la durata attiva di un processo.

=.=.=.=.

#### DESCRIZIONE

La presente invenzione riguarda un procedimento per introdurre variazioni in un sistema controllato dal programma memorizzato, senza richiedere arresto parziale o totale del sistema e successiva inizializzazione.

#### Base dell'invenzione

I sistemi controllati da programma sono stati applicati ad un'ampia gamma di applicazioni di elaborazione e controllo di dati. La flessibilità funzionale di questi sistemi deriva dal programma, memorizzato nella memoria di questi sistemi, che controlla il loro funzionamento.

In sistemi o impianti controllati da programma, è frequentemente necessario introdurre variazioni di programma dopo l'installazione iniziale per l'adattamento di nuovi servizi e/o nuovo hardware. Il modo più comune per introdurre questi cambiamenti è quello di arrestare il sistema e inizializzarlo completamente. Sfortunatamente, ciò rende il sistema incapace di fornire servizio per l'intervallo di inizializzazione, solitamente un periodo di vari minuti o anche ore. In aggiunta, l'inizializzazione distrugge frequentemente alcune porzioni di dati di più lungo termine che possono influenzare le prestazioni successive del sistema; ad esempio, essa può azzerare dati forniti da un utente telefonico definenti un certo aspetto del proprio servizio. L'inizializzazione è particolarmente indesiderabile se il sistema controllato da programma controlla processi continui, come ad esempio processi chimici; piani accurati devono essere ese-

guiti per introdurre tale inizializzazione in corrispondenza di un momento in cui i processi chimici controllati sono stati tutti arrestati o hanno raggiunto un certo equilibrio.

Un metodo meno drastico per introdurre cambiamenti di programma con inizializzazione parziale è attuato, ad esempio, in impianti di commutazione telefonica, come ad esempio l'Impianto di Commutazione Elettronico No. 1 descritto nel Bell System Technical Journal, Settembre, Edizione 1964 e nel brevetto statunitense N. 3.570.008. Le caratteristiche speciali di questi sistemi hanno reso possibile introdurre variazioni senza disturbare quegli utenti che sono già collegati ad altri utenti. Tutti i dati relativi a connessioni completate e a servizi di utente sono isolati e trattati in modo speciale tramite programmi di inizializzazione. Questi sistemi hanno processori duplicati, uno dei quali è del tipo "in linea" e l'altro dei quali è un'unità di riserva pronta per assumere il controllo. Una versione modificata di un programma viene caricata in uno dei processori, mentre il processore si trova nel modo a riposo. Esso viene quindi commutato "in linea" e esegue un programma di inizializzazione che non modifica nessuno dei dati associati con connessioni completate o servizi di utente. La versione modificata del programma è quindi pure caricata nell'altro processore, ed il sistema si trova in uno stato stabile. Utenti che stavano eseguendo una connessione non avranno le loro chiamate completate, e perciò questo tipo di cambiamento viene attuato durante ore di attività molto ridotta dell'impianto o sistema per minimizzare i disturbi. Questo metodo di introdurre variazioni non è applicabile ai sistemi controllati da programma moderni più sofisticati. Questi sistemi

sono normalmente fatti funzionare sotto il controllo di un programma complesso chiamato "sistema operativo" che avvia altri programmi chiamati "processi". L'esecuzione effettiva dei compiti del sistema o dell'impianto è normalmente attuata tramite i processi tramite l'esecuzione di una serie di segmenti di programma chiamati "funzioni" di programma. L'esecuzione delle funzioni di programma può essere avviata mediante una chiamata di funzione da un qualsiasi processo nel sistema. Le funzioni di programma accedono ad una base di dati comune, grandi porzioni della quale sono controllate da molti processi, in cui i concatenamenti fra i vari elementi di dati associati sono molteplici e complessi. Queste strutture di dati complesse rendono drasticamente complicato il tipo di trattamento speciale di una porzione selezionata di dati impiegata nei sistemi della tecnica nota summenzionati. In tal modo, l'inizializzazione di taluni processi e nel contempo la ritenzione di altri, mantenendo al tempo stesso una base di dati coerente, è estremamente difficoltosa. E' inoltre impossibile prevedere la natura delle variazioni del programma e pianificare queste variazioni nella progettazione della base di dati. Per tutte le ragioni precedenti, si è rivelato difficile introdurre tutte le variazioni di programma, tranne quelle secondarie, nei sistemi o impianti controllati da programma moderni più sofisticati senza inizializzazione massiccia.

~~Questo problema è stato risolto dalla presente invenzione, il cui metodo comprende le fasi di generare un numero sequenziale di sistema che è variato solo in modo monotono in una data direzione, assegnare il numero sequenziale di sistema ad un processo con un numero sequenziale di processo in corrispondenza del momento in cui il processo viene avviato,~~

memorizzare una versione modificata di una funzione di programma selezionata nella memoria del programma, ritenendo al tempo stesso una versione attuale della funzione di programma selezionata, registrare un numero di sequenza di funzione decisionale, il numero di sequenza di funzione decisionale essendo oltre il numero sequenziale o di sequenza del sistema nella data direzione o dato senso, variare il numero sequenziale del sistema nella data direzione in modo tale che esso abbia a raggiungere un valore progredito almeno nella data direzione come il numero sequenziale di funzione decisionale, e eseguire la versione attuale se il numero di sequenza o sequenziale di processo è prima del numero sequenziale di funzione decisionale nella data direzione, e eseguire la versione modificata se il numero sequenziale di processo è almeno progredito come il numero sequenziale di funzione decisionale nella data direzione.

#### Sommario dell'invenzione

Uno scopo della presente invenzione è quello di consentire la introduzione di variazioni di programma in sistemi controllati da programma senza interferire con i processi in corso di esecuzione, anche mentre il sistema sta funzionando con carico normale.

Secondo la presente invenzione, cambiamenti sono introdotti nel programma di un sistema o impianto controllato da programma memorizzando una versione modificata di una funzione di programma, ritenendo nel contempo la versione <sup>corrente</sup> attuale e facendo quindi sì che tutti i processi avviati prima della variazione abbiano ad eseguire la versione <sup>corrente</sup> attuale di tale funzione, e facendo in modo che tutti i processi avviati dopo la variazione abbiano ad eseguire la versione modificata di detta funzione.

Secondo una forma di realizzazione, ciò viene attuato con

l'ausilio di un numero sequenziale di sistema, mantenuto dal sistema; Un numero sequenziale di sistema viene variato solo in modo monotono in una direzione o dato senso, e viene variato almeno ogni volta che una funzione di programma modificata viene introdotta nel sistema. Il valore <sup>corrente</sup> ~~attuale~~ del numero sequenziale del sistema viene registrato con un processo ogniqualvolta il processo viene avviato. Ogniqualvolta una versione modificata di una funzione di programma viene memorizzata nel sistema, un numero sequenziale di funzione decisionale viene assegnato alla funzione modificata, avente un valore il quale è oltre nella data direzione, rispetto al valore del numero sequenziale di processo di un qualsiasi processo <sup>correntemente</sup> ~~attual-~~ mente attivo che impiega tale funzione di programma. Il numero sequenziale di sistema viene successivamente modificato, in modo tale che esso abbia a trovarsi nella data direzione almeno progredito come il numero sequenziale di funzione decisionale. Successivamente, ogniqualvolta un processo richiede la funzione che è stata modificata, il numero sequenziale di funzione decisionale viene comparato con il numero di processo. Se il risultato della comparazione indica che la versione modificata è stata memorizzata nel sistema prima dell'inizio del processo di chiamata, allora viene eseguita la versione modificata della funzione di programma; in caso contrario viene eseguita la funzione <sup>corrente</sup> ~~attuale~~.

In una forma di realizzazione specifica dell'invenzione, funzioni di programma possono essere richieste da un processo tramite l'impiego di un vettore in una tabella vettoriale di trasferimento memorizzata, che normalmente punta direttamente alla funzione di programma desiderata. Se viene introdotta una funzione di programma modificata nel sistema, al-

lora il vettore, invece punterà ad una funzione decisionale. La funzione decisionale esegue quindi la decisione relativa al fatto se deve essere eseguita la versione <sup>corrente</sup> ~~attuale~~ oppure la versione modificata della funzione di programma. Se deve essere eseguita la versione <sup>corrente</sup> ~~attuale~~, allora viene eseguita una diramazione di programma alla versione <sup>corrente</sup> ~~attuale~~; in caso contrario viene eseguita una diramazione di programma alla versione modificata. Vantaggiosamente, la funzione decisionale può precedere direttamente la versione modificata, semplificando così il processo di recupero di spazio di memorizzazione quando versioni vecchie di funzione di programma non sono più necessarie.

Periodicamente, un programma di audit può essere eseguito per determinare se un qualsiasi processo attivo riguardante una funzione di programma, che è stata modificata, ha un numero sequenziale che è, nella data direzione, prima di quello della versione modificata della funzione di programma. Se non vi sono processi di questo tipo, allora la versione <sup>corrente</sup> ~~attuale~~ della funzione di programma può essere cancellata, il vettore può puntare direttamente alla versione modificata della funzione invece di puntare alla funzione decisionale, e il blocco di memoria occupato dalla versione <sup>corrente</sup> ~~attuale~~ della funzione di programma diviene spazio disponibile per altre funzioni di programma nuove o modificate.

Vantaggiosamente, i numeri sequenziali possono essere costituiti da qualsiasi numero a gruppo di numeri che variano in modo monotono, o tramite incrementazione, oppure tramite decrementazione, per cui è possibile indicare se un dato numero sequenziale era assegnato prima di un altro numero sequenziale. Numeri sequenziali devono variare con una fre-

quenza almeno pari a quella con cui sono introdotte le variazioni di programma e possono variare molto più frequentemente. Ad esempio, è possibile impiegare una quantità rappresentante data e ora come un numero sequenziale, poichè questa quantità si mantiene variabile in modo monotono tramite incrementazione positiva. Ciò presenta l'ulteriore vantaggio di impiegare un numero sequenziale che può avere impiego aggiuntivo, ad esempio per rivelare processi che sono stati in stato attivo per un periodo di tempo eccessivo. Inoltre, se è desiderato caricare una modifica un certo tempo prima che la modifica debba essere attivata, allora il numero sequenziale di funzione decisionale può essere preimpostato ad una quantità rappresentante il tempo di attivazione; la modifica non sarà in tal caso impiegata prima del citato tempo di attivazione.

Breve descrizione del disegno

L'invenzione sarà meglio compresa dalla seguente descrizione particolareggiata, quando letta in unione ai disegni in cui:

la figura 1 è una disposizione di memoria di una forma di realizzazione illustrativa dell'invenzione, rappresentante le relazioni di indirizzamento fra versioni <sup>coerente</sup> attuate e modificata di una funzione di programma;

la figura 2 è uno schema a blocchi di un processore di un sistema o impianto controllato da programma; e

la figura 3 è un diagramma di flusso di un programma di audit per recuperare memorizzazione di una versione vecchia di un programma quando non più richiesta.

Descrizione dettagliata

Un sistema o impianto controllato da programma ha due tipi di informazione memorizzata nella sua memoria e cioè: programmi e dati. Un

programma comprende un insieme di istruzioni basilari di un processore, ciascuna delle quali istruzioni dirige una certa fase elementare, per cui un insieme quando eseguito realizza un qualche compito più ampio del sistema o serie di compiti. I dati comprendono informazione relativa allo stato <sup>corrente</sup> ~~attuale~~ di un compito, informazione ottenuta da sorgenti esterne e preliminarmente o precedentemente memorizzata nel sistema, e informazione generata per eseguire i compiti funzionali del sistema. I programmi eseguono i loro compiti elaborando dati e controllando sistemi di ingresso/uscita in conformità con i valori attuali dei dati.

In sistemi controllati da programma moderni, il controllo basilare del sistema risiede in un programma principale il quale, assieme ai dati associati, viene chiamato il sistema operativo. Il sistema operativo dirige l'esecuzione dei compiti del sistema creando e <sup>determinando</sup> ~~attuando~~ processi. Un processo è un programma più memoria di dati di controllo associata, chiamato blocco di controllo di processo. Il blocco di controllo di processo mantiene traccia o registrazione del contesto di ciascun processo includente informazione come ad esempio l'identificazione del processo, lo stato corrente del processo, la priorità di processo, il tempo del sistema in corrispondenza dell'inizio del processo, ecc. I processi, a loro volta, eseguono i loro compiti richiedendo l'esecuzione di una serie di funzioni di programma.

Funzionando normalmente, un sistema controllato da programma avrà solitamente alcuni processi che sono attivi e alcuni che sono a riposo. Un processo è considerato come attivo durante il periodo in cui esso sta attivamente eseguendo un compito, attendendo un blocco o periodo di tempo

per la continuazione della propria esecuzione di un compito, o attendendo dispositivi di ingresso oppure di uscita. In seguito al completamento di un compito, o gruppo di compiti, esso assume lo stato di riposo e rimane in questo stato finchè esso non viene nuovamente avviato per eseguire un altro compito o gruppo di compiti. Quando un processo entra in uno stato di riposo, un minimo di dati viene riportato per l'impiego successivo al riavviamento del processo. Nessun riferimento di programma diretto è ritenuto quando un processo passa a riposo.

In corrispondenza di uno stadio qualsiasi nella vita di un processo, una variazione di programma può essere introdotta memorizzando una funzione di programma modificata in un certo spazio disponibile in memoria. Tuttavia, risultati non corretti possono essere generati se una funzione di programma modificata viene impiegata da un processo che era già attivo prima della introduzione della variazione. Al fine di garantire che ciascun processo abbia ad impiegare la medesima funzione di programma durante tutto il tempo in cui esso rimane attivo, funzioni decisionali sono associate con ciascuna versione modificata di una funzione di programma. Le funzioni decisionali sono brevi programmi che determinano quando il processo che ha richiesto la funzione di programma è stato iniziato - prima o dopo del caricamento nel sistema o impianto della versione modificata della funzione di programma. Se la funzione modificata è stata caricata prima dell'avviamento del processo, allora viene impiegata la versione modificata stessa; se la versione modificata è stata caricata dopo l'avviamento del processo, allora viene impiegata la versione <sup>corrente</sup> ~~attuale~~.

Un numero sequenziale del sistema è mantenuto per tener trac-

cia e registrazione dell'ordine di eventi, cioè se un processo è stato iniziato prima o dopo il caricamento nel sistema di una versione modificata di una funzione di programma. Un numero sequenziale di sistema può essere costituito da una qualsiasi quantità variata in modo monotono, in modo tale che può essere stabilito l'ordinamento nel tempo tra due numeri sequenziali diversi. Ad esempio, se un numero sequenziale di sistema è sempre incrementato in modo positivo, allora un numero sequenziale di sistema maggiore è associato con un istante temporale maggiormente posteriore che non un numero sequenziale di sistema inferiore. Il tempo, comprendente data e tempo del giorno viene impiegato come un numero sequenziale di sistema in questa forma di realizzazione.

Nella figura 1, il blocco 100 contiene il numero sequenziale del sistema, il tempo, aggiornato automaticamente dal sistema operativo impiegando l'orologio del sistema. L'orologio del sistema costituisce parte del processore 200. Esso è ben noto nella tecnica e non è rappresentato nel disegno. Il meccanismo per aggiornare una entrata temporale in memoria è pure noto, e non sarà ulteriormente descritto nel presente contesto.

Un numero sequenziale di processo rappresentante tempo viene memorizzato nel blocco di controllo di processo di un processo ogniqualvolta esso viene avviato. Il blocco 110 di controllo di processo contiene l'informazione di controllo di processo associata con un processo tipico. Come si è detto precedentemente, il blocco di controllo di processo include vari elementi di informazione associati col processo. La posizione 111 del blocco 110, contiene il numero sequenziale di processo

che indica il momento in corrispondenza del quale è stato avviato il processo associato. Ogniqualvolta un processo diviene attivo, il valore ~~attuale~~ <sup>corrente</sup> del numero sequenziale del sistema viene posto nella posizione 111.

Un numero sequenziale di funzione decisionale, rappresentante il tempo <sup>Cox -</sup> ~~attuale~~ <sup>corrente</sup> incrementato di un secondo, viene assegnato alla funzione decisionale in corrispondenza del momento in cui una funzione di programma modificata viene caricata nella memoria.

Ciascuno dei blocchi funzionali 130, 140 e 150 di figura 1 rappresenta un'area di memoria per memorizzare un programma. Ciascuno ha un'area per memorizzare una funzione decisionale (131, 141 e 151) e un'area per memorizzare una funzione di programma (132, 142 e 152). Le funzioni di programma rappresentano una versione <sup>corrente</sup> ~~attuale~~,  $F$ , (132), una versione modificata,  $F'$ , (142) ed una seconda modifica,  $F''$  (152). Il blocco 141 di funzione decisionale comprende un programma 144 di funzione decisionale e un numero sequenziale 143 di funzione decisionale identificato come  $m$ . Il programma di funzione decisionale può essere rappresentato come segue: se il numero sequenziale di processo,  $n(pc)$ , è inferiore al numero sequenziale di funzione decisionale,  $m$ , allora il programma passa a  $f(2)$  (blocco 132); in caso contrario va a  $f'(2)$  (blocco 142). In altre parole, se il processo è stato avviato prima della introduzione della variazione, allora viene impiegata la versione attuale della funzione di programma,  $F$ ; in caso contrario impiega la versione modificata della funzione di programma,  $F'$ .

I programmi sono assegnati a qualsiasi segmento disponibile di memoria. Funzioni di programma individuali possono essere impiegate da

molti processi, ma memorizzate solo una volta per risparmiare spazio di memoria utile. Poichè molti programmi possono richiamare una funzione di programma, è mantenuta una lista degli indirizzi <sup>correct</sup> attuali di tutte le funzioni di programma ed essa è resa disponibile per tutti i programmi chiamanti. Questa lista viene chiamata tabella vettoriale di trasferimento. Quando una variazione viene caricata nel sistema, l'indirizzo della tabella vettoriale di trasferimento viene aggiornato per puntare alla funzione decisionale associata con la versione modificata. Quando un processo richiede la funzione di programma che è stata modificata, la funzione decisionale puntata dalla tabella vettoriale di trasferimento determinerà se deve essere impiegata la versione <sup>correct</sup> attuale o modificata della funzione di programma. Nella tabella vettoriale di trasferimento 120, l'entrata 121 punta all'indirizzo della funzione di programma o funzione decisionale esemplificative. Se solo una versione di un programma viene caricata nel sistema, allora l'entrata 121 rappresenta l'indirizzo della funzione di programma. Se più di una versione viene caricata nel sistema, allora l'entrata 121 rappresenta l'indirizzo di una funzione decisionale che consentirà la selezione della versione corretta del programma. Quando la versione modificata della funzione di programma  $F'$  viene caricata nella memoria del sistema, l'entrata della tabella vettoriale in corrispondenza di 121 viene variata da  $f(2)$  che è l'indirizzo della versione attuale della funzione di programma, a  $f'(1)$ , che è l'indirizzo della funzione decisionale associata con  $F'$ . Come risultato, prima che sia eseguita l'una o l'altra versione della funzione di programma, la funzione decisionale compara i numeri sequenziali di processo e funzione decisionale per decidere se il pro-

cesso è stato iniziato prima o dopo l'introduzione della variazione. Se il processo è stato avviato prima della variazione, allora si verifica una diramazione alla versione attuale  $F$ ; in caso contrario viene eseguita la versione modificata  $F'$ .

Se sussiste una qualche necessità di avere una modifica addizionale, allora deve essere introdotta una seconda modifica rappresentata da  $F''$  nel blocco 150. Quando  $F''$  viene caricata nel sistema, l'entrata 121 nella tabella vettoriale di trasferimento viene variata per puntare a  $f''(1)$ , che è la funzione decisionale associata con  $F''$ . Il blocco 150 comprende la funzione decisionale, blocco 151, e la funzione di programma,  $F''$ , blocco 152. Il blocco della funzione decisionale comprende il numero sequenziale 153 di funzione decisionale, rappresentato da  $p$ , ed il programma 154 di funzione decisionale. Il programma 154 di funzione decisionale può essere dichiarato come segue: se il numero sequenziale di processo  $n(pc)$ , è inferiore al numero sequenziale di funzione decisionale,  $p$ , allora va a  $f'(1)$ ; in caso contrario va a  $f''(2)$ . In altre parole, se il processo è stato avviato prima del caricamento di  $F''$ , allora viene eseguito un salto al programma 144 di funzione decisionale associato con la versione  $F'$ ; in caso contrario viene eseguita  $F''$ . Il risultato del programma 144 di funzione decisionale è un salto alla funzione originale,  $F$ , se il numero sequenziale di processo,  $n(pc)$  è inferiore al numero sequenziale di funzione decisionale  $m$ ; in caso contrario viene eseguita la funzione di programma  $F'$ . In questo esempio, i processi impiegano le funzioni di programma  $F$ ,  $F'$ , o  $F''$ . I processi più vecchi impiegano  $F$ , i processi intermedi impiegano  $F'$  e i processi avviati più recentemente impiegano  $F''$ . Se è necessario o

desiderato modificare simultaneamente un gruppo di funzioni, allora una funzione decisionale viene inserita per ciascuna di queste funzioni di programma, e il medesimo numero sequenziale viene assegnato a tutte le funzioni decisionali associate con tale gruppo. E' pure possibile attivazione ritardata di una modifica di programma. Il numero sequenziale di funzione decisionale associato con tale modifica sarebbe il tempo di attivazione ritardato. La modifica non avrebbe quindi effetto sino al tempo di attivazione.

Questa descrizione è stata fornita nei termini di una modifica di una funzione di programma singola. Manifestamente, molte funzioni di programma possono essere modificate impiegando le tecniche della presente invenzione. Modificazioni di programma per molte funzioni di programma diverse, caricate in corrispondenza di momenti diversi, ciascuna funzione di programma modificata avendo la sua propria funzione decisionale associata, possono esistere contemporaneamente nel sistema.

La figura 2 illustra tre componenti principali di un sistema controllato da programma, e cioè il processore centrale 202, la memoria principale 201 e l'apparecchiatura di ingresso/uscita 203. L'unità di elab-

*Per esempio, il VAX 11/780, fabbricato dalla Digital Equipment Corporation, include la memoria associata, e un dispositivo di trattamento simile.*

borazione centrale e la sua memoria associata possono essere uno qualsiasi di una pluralità di moderni processori ben noti nella tecnica. L'apparecchiatura di ingresso/uscita può includere dispositivi di memoria supplementare come ad esempio unità a nastro magnetico e dischi a testine fisse oppure mobili. L'unità di elaborazione centrale esegue processi e funzioni di programma che sono memorizzati in memoria. Nell'eseguire questi processi e funzioni di programma, l'unità di elaborazione centrale accede e mo-

difica dati memorizzati nella memoria. L'unità di elaborazione centrale esegue semplici passi come ad esempio la comparazione di due numeri, l'esecuzione di un ramo condizionato basato su tale comparazione, l'incrementazione di una quantità nella memoria, lo spostamento di quantità memorizzate in memoria da una posizione all'altra, la diramazione incondizionata ad un indirizzo indicato direttamente oppure indirettamente. L'unità di elaborazione centrale è pure in grado di caricare una versione modificata di una funzione di programma dalla apparecchiatura di ingresso/uscita 203 nella memoria principale 201.

Periodicamente, vengono eseguiti programmi di audit sia per recuperare spazio di memoria occupato da versioni vecchie delle funzioni di programma che non sono più impiegate sia per aggiornare la tabella vettoriale di trasferimento con indirizzi di versioni modificate di funzioni di programma, al fine di bypassare la funzione decisionale non più richiesta. La figura 3 è un diagramma di flusso di un programma di audit che esamina processi che potrebbero impiegare o la funzione di programma vecchia  $F$  oppure la versione modificata  $F'$ . In questo esempio si suppone che il programma di audit sia eseguito dopo l'introduzione della modifica  $F'$  nel sistema e prima che abbia a verificarsi la modifica  $F''$ .

Il programma di audit può essere considerato come suddiviso in tre parti e cioè: passi iniziali (300-302), ciclo (303-306) e recupero memoria (307-308). Il programma di audit viene avviato tramite il sistema operativo. Il programma di audit deve leggere il numero sequenziale  $m$  di funzione decisionale che deve essere comparato col numero sequenziale di processo di tutti i processi attivi che potrebbero impiegare la versione at-

tuale o modificata della funzione di programma. Viene quindi fatto accesso al primo processo. La serie di passi del ciclo inizia con un controllo per determinare se il processo è in stato di riposo o inattivo. Se il processo non è in stato inattivo, allora il numero sequenziale di processo viene comparato col numero sequenziale di funzione decisionale. Se il numero sequenziale di processo non è almeno pari al numero sequenziale di funzione decisionale, allora il programma di audit viene terminato; ciò significa che un qualche processo attivo sta ancora impiegando la versione attuale della funzione di programma. In caso contrario, il processo ciclico continua finchè tutti i processi appropriati non sono stati esaminati. Se il programma di audit non è stato terminato dopo che tutti i processi appropriati sono stati esaminati, allora il programma di audit può procedere per recuperare memoria. Esso fa ciò variando l'entrata della tabella vettoriale in modo che punto direttamente alla versione modificata, e rendendo disponibile lo spazio di memoria occupato dalla versione corrente per il sistema operativo per la riassegnazione per altri scopi.

Come è illustrato nel blocco 300 del diagramma di flusso di figura 3, il programma di audit inizia l'esecuzione dopo essere stato avviato dal sistema operativo. Il numero sequenziale di funzione decisionale,  $m$ , associato con  $F'$ , viene letto dal programma di audit in 301. In 302, il programma di audit accede al primo processo che impiega  $F$  o  $F'$ . Quindi, in corrispondenza dei punti decisionali 303 e 304, rispettivamente, il programma di audit determina se il processo è inattivo e se il suo numero sequenziale,  $n(pc)$ , è maggiore del o uguale al numero sequenziale di funzione decisionale associato con  $F'$ . Se entrambe queste decisioni sono negative,

allora il programma di audit conclude che il processo è attivo e impiega F, la versione <sup>corrente</sup> ~~attuale~~ della funzione di programma. Poichè F è ancora impiegata da un processo attivo, essa non può essere cancellata. Conseguentemente, il programma di audit termina in corrispondenza di 309.

Se l'una o l'altra decisione è positiva, allora il programma di audit conclude che il processo è o a riposo o inattivo e preparato per impiegare la versione modificata della funzione di programma F' quando essa viene avviata, oppure sta già impiegando F'. Il programma di audit procede al punto decisionale 305 ove esso determina se vi sono ulteriori processi che impiegano F o F'. Se vi sono altri di questi processi, allora il programma di audit accede al processo successivo nel blocco 306, e cicla attraverso i punti decisionali 303, 304 e 305 nuovamente.

Il programma di audit ripete il ciclo finchè tutti i processi che impiegano le funzioni F o F' non sono stati esaminati. Se non sono stati scoperti processi attivi impieganti F, allora non vi è ragione di conservare l'indirizzo della funzione decisionale nella tabella vettoriale di trasferimento. Perciò, nel blocco 307, il programma di audit cambia l'entrata 121 nella tabella di trasferimento per puntare direttamente all'indirizzo f'(2) della nuova funzione di programma F'.

Poichè la funzione di programma F non è più puntata da alcun processo e non può più essere indirizzata dalla funzione decisionale, essa occupa spazio sprecato in memoria. Poichè questo spazio può essere impiegato per un qualche altro scopo, il programma di audit cancella la funzione di programma F, rilascia il blocco 130 come uno spazio di memoria libero in corrispondenza di 308 e termina in corrispondenza di 309. Si deve no-

tare che il blocco 130 include pure lo spazio precedentemente occupato da una funzione decisionale quando F è stata introdotta come una modifica del programma.

Ritendendo o assicura la modifica e la sua funzione decisionale associata, l'interblocco di memoria può essere reso disponibile quando F non è più richiesta.

In taluni sistemi può essere difficile che il programma di audit possa determinare se un processo è attivo o inattivo. In questo caso, il controllo del blocco 303 può essere bypassato. Può pure essere più conveniente esaminare tutti i processi indipendentemente dal fatto se essi impiegano o no F o F'. Gli unici risultati nocivi di tale semplificazione sono che la funzione decisionale rimane attiva per un periodo maggiore di quello necessario, e che lo spazio di memorizzazione occupata dalla versione attuale della funzione di programma non può essere altrettanto rapidamente recuperato.

Una grande classe di modifiche di programma possono essere introdotte impiegando le tecniche della presente invenzione, fra cui modifiche in programmi che avviano e eseguono processi. In linea generale, questi programmi sono funzioni di programma richieste dal sistema operativo che è pure un gruppo di funzioni di programma. Qualsiasi programma richiesto indirettamente può in linea di principio, essere variato impiegando le tecniche della presente invenzione. Tuttavia, talune restrizioni sorgono nell'eseguire modificazioni a funzioni di programma che impiegano dati interpretati differentemente dalle versioni attuale e modificata di una funzione di programma, in particolare quando questi dati sono conservati durante un intervallo inattivo di un processo oppure quando questi dati sono

condivisi da due o più processi. Anche quando si accede a questi dati, un impiego accurato di funzioni decisionali per dati, congiuntamente a funzioni decisionali per funzioni di programma, può consentire l'introduzione di modifiche in questi dati impiegando le tecniche della presente invenzione. Si deve notare che un prerequisito implicito è un unico punto di accesso come ad esempio un vettore di trasferimento per funzione di programma, impiegato ogniqualvolta una funzione decisionale viene impiegata per accedere a questi dati.

Si deve tener presente che la forma di realizzazione precedentemente descritta è semplicemente illustrativa dei principi della presente invenzione; altre disposizioni possono peraltro essere concepite dagli esperti del ramo senza allontanarsi dallo spirito e ambito protettivo dell'invenzione.

Ciò che si rivendica è:

#### ~~RIVENDICAZIONI~~

1. In un impianto di dispositivo di trattamento controllato o  
~~1. Procedimento per introdurre variazioni in un sistema o in~~  
~~programma in cui è iniziato un processo ed è eseguito in funzione e chiuso~~  
~~pianto controllato da programma senza richiedere arresto parziale o tota-~~  
~~le funzioni di programma memorizzate in una memoria di programma~~  
~~un procedimento di modificazione di dette funzioni di programma~~  
~~le del sistema e successiva inizializzazione, comprendente le fasi di~~  
~~durante l'esecuzione di detto procedimento, detto metodo comprendente~~  
la fase di generare un numero sequenziale di sistema il quale viene va-

riato solo in modo monotono in una data direzione o in un dato senso;

assegnare il numero sequenziale di sistema <sup>a detto</sup> ad un processo co-

me un numero sequenziale di processo in corrispondenza del momento in cui  
<sup>detto</sup>  
il processo viene avviato;

memorizzare una versione modificata di una <sup>di detto</sup> funzione di pro-  
gramma selezionata <sup>in detto</sup> nella memoria di programma, conservando al tempo stes-  
<sup>corrente di detto</sup>  
so una versione ~~attuale della~~ funzione di programma selezionata;

registrare un numero sequenziale di funzione decisionale,  
~~il~~ <sup>detto</sup> numero sequenziale di funzione decisionale essendo oltre ~~il~~ <sup>detto</sup> numero sequenziale di <sup>processo</sup> ~~sistema o dell'impianto~~ <sup>in detto</sup> nella data direzione;  
variare il numero sequenziale del sistema ~~nella~~ <sup>in detto</sup> data direzione in maniera tale che esso abbia a raggiungere un valore almeno progredito <sup>in detto</sup> nella data direzione come il numero sequenziale di funzione decisionale; e <sup>in risposta od un richiamo di funzione per detta funzione scelta,</sup> eseguire la versione <sup>detto</sup> ~~attuale~~ <sup>corrente</sup> se il numero sequenziale di processo precede il numero sequenziale di <sup>in detto</sup> ~~funzione~~ <sup>detto</sup> decisionale <sup>detto</sup> nella data direzione, e eseguire <sup>detto</sup> ~~la~~ <sup>detto</sup> versione modificata se ~~il~~ <sup>detto</sup> numero sequenziale di processo è almeno progredito come ~~il~~ <sup>detto</sup> numero sequenziale di funzione decisionale <sup>in detto</sup> nella data direzione.

2. Procedimento secondo la rivendicazione 1, in cui ~~la~~ <sup>detta</sup> fase di esecuzione comprende inoltre le fasi di:  
comparare <sup>detto</sup> ~~il~~ numero sequenziale di processo con <sup>detto</sup> numero sequenziale di funzione decisionale;  
diramare <sup>a detto</sup> ~~alla~~ <sup>corrente</sup> versione <sup>di detto</sup> ~~attuale~~ se il risultato della comparazione indica che <sup>detto</sup> ~~il~~ numero sequenziale di processo si trova prima <sup>di detto</sup> del numero di funzione <sup>sequenziale in detto</sup> ~~decisionale~~ nella data direzione; e  
diramare <sup>a detto</sup> ~~alla~~ <sup>di detto</sup> versione modificata se il risultato della comparazione indica che <sup>detto</sup> ~~il~~ numero sequenziale di processo si trova progredito almeno come <sup>in detto</sup> ~~il~~ numero sequenziale di funzione decisionale nella data direzione.

3. Procedimento secondo la rivendicazione 1, in cui ~~la~~ <sup>detta</sup> fase di assegnazione comprende la fase di:  
memorizzare <sup>detto</sup> ~~il~~ numero sequenziale del sistema o dell'impianto

in una posizione di memoria associata con <sup>del</sup> il processo.

4. Procedimento secondo la rivendicazione 1, in cui detta versione <sup>corrente</sup> attuale di detta funzione di programma selezionata comprende una versione intermedia ed una versione più vecchia di detta funzione di <sup>del</sup> programma selezionato, e in cui detta fase di diramazione a detta versione <sup>corrente</sup> attuale, se risultato di detta comparazione, indica che detto numero sequenziale di processo è prima di detto numero sequenziale di funzione decisionale in detta data direzione, comprende inoltre le fasi di:

eseguire una seconda comparazione fra detto numero sequenziale di processo e un secondo numero sequenziale di funzione decisionale associato con detta versione intermedia di detta funzione di programma selezionata;

diramare a detta versione più vecchia se detta seconda comparazione indica che detto numero sequenziale di processo si trova prima di detto secondo numero sequenziale di funzione decisionale in detta data direzione; e

diramare a detta versione intermedia se detta seconda comparazione indica che detto numero sequenziale di processo è almeno progredito come detto secondo numero sequenziale di funzione decisionale in detta data direzione.

5. Procedimento secondo le rivendicazioni 1, 2, 3 oppure 4 comprendente inoltre le fasi di:

ripetere detta fase di assegnazione per una pluralità di processi, compresi tutti i processi che realizzano chiamate funzionali a detta funzione di programma selezionata;

eseguire periodicamente terze comparazioni fra il numero sequenziale di processo di almeno tutti i processi attivi inclusi in detta pluralità di processi e detto numero sequenziale di funzione decisionale; e cancellare detta versione <sup>corrente</sup> attuale se nessuno di detti numeri sequenziali di processo di almeno tutti i processi attivi inclusi in detta pluralità di processi si trova prima di detto numero sequenziale di funzione decisionale in detta data direzione.

6. Sistema a processore controllato da programma comprendente:

un'unità di elaborazione centrale (202) ed una memoria di programma (201) caratterizzato dal fatto che la memoria di programma (201) contiene una versione attuale di una funzione di programma selezionata (132); e memorizza una versione modificata della funzione di programma selezionata (142);

l'unità di elaborazione centrale (202) interroga la memoria di programma (201, 110, 111, 121, 141, 132, 142) e esegue la versione attuale di una funzione di programma selezionata se un processo richiedente la funzione di programma selezionata è stato avviato prima della memorizzazione della versione modificata nella memoria, e esegue la versione modificata se il processo richiedente la funzione di programma selezionata è stato iniziato dopo la memorizzazione in memoria della versione modificata.

7. Sistema o impianto a processore controllato da programma secondo la rivendicazione 6, caratterizzato dal fatto che:

è memorizzato un numero sequenziale del sistema, il quale numero viene variato in un dato senso o data direzione;

il numero sequenziale del sistema è memorizzato come un nume-

~~in risposta ad un richiamo di funzione per detta funzione  
scelta~~

6. Il procedimento secondo la rivendicazione 2 in cui detto impianto comprende almeno una tavola di vettore comprendente una serie di indici per funzioni di programma ed in cui detta fase di variare un indice in una di dette tavole di vettori che indica a detta versione corrente di detta funzione di programma scelta per indicare un programma per eseguire dette fasi di comparazione e di diramazione.

7. Il procedimento secondo le rivendicazioni 1, 2, 3, 4, o 6 in cui detto numero di sequenza dell'impianto è il tempo del giorno e della data.

8. In un impianto di trattamento di programma controllato, in cui un procedimento è iniziato ed eseguito in funzione, richiama le funzioni di programma memorizzate in una memoria di programma, mezzi per modificare funzioni di programma durante l'esecuzione di detto procedimento, detti mezzi comprendenti:

il mezzo per memorizzare un numero di sequenza d'impianto che è cambiato solo in una data direzione;

il mezzo per memorizzare detto numero di sequenza d'impianto al momento in cui è iniziato detto procedimento;

mezzo per memorizzare una versione modificata di una di quelle scelte di dette funzioni di programma in detta memoria di programma mentre si conserva una versione corrente di detta funzione scelta di programma in detta memoria;

mezzo per memorizzare un numero di sequenza di funzione di decisione al momento di memorizzare detta versione modificata, detto numero di sequenza di funzione di decisione essendo oltre detto numero di sequenza di procedimento in detta direzione data;

mezzo per modificare detto numero di sequenza del sistema in modo che esso raggiunga un valore almeno progredito in detta direzione data come detto numero sequenziale di funzione decisionale;

mezzo per eseguire detta versione modificata di detta funzione scelta in risposta ad una richiesta di funzione per detta funzione scelta se detto numero sequenziale di processo è prima di detto numero sequenziale di funzione decisionale in detta direzione data; e mezzo per eseguire detta versione corrente di detta funzione scelta in risposta ad un richiamo di funzione per detta funzione scelta se detto numero sequenziale di funzione decisionale è almeno fino a detto numero sequenziale di funzione decisionale in detta direzione data.

9. Un sistema di processore a programma controllato

secondo la rivendicazione 8 in cui detto mezzo per l'esecuzione comprende inoltre:

il mezzo per confrontare detto numero sequenziale di processo con detto numero sequenziale di funzione decisionale:

il mezzo per diramare detta versione corrente se detto confronto indica che detto numero sequenziale di processo è prima di detto numero sequenziale di funzione decisionale in detta direzione data;

e mezzo per diramare a detta versione modificata se detto confronto indica che detto numero sequenziale di processo è almeno fino a detto numero sequenziale di funzione decisionale in detta direzione data.

10. Sistema o impianto a processore controllato da programma secondo la rivendicazione 9 in cui detta versione corrente di detta funzione di programma scelta comprende una versione intermedia ed una versione più vecchia di detta funzione di programma scelta, ed in cui detto mezzo per diramare a detta versione corrente se detto confronto indica che detto numero sequenziale di processo è prima di detto numero sequenziale di funzione decisionale in detta direzione data, comprende inoltre, :

il mezzo per fare un secondo confronto tra detto numero sequenziale di funzione decisionale associato con detta

versione intermedia di detta funzione di programma scelta;

mezzo per diramare a detta versione più vecchia se detto confronto secondario indica che detto numero sequenziale di processo è prima di detto numero sequenziale di funzione decisionale in detta direzione data;

e mezzo per diramare a detta versione intermedia se detto secondo confronto indica che detto numero sequenziale di processo è almeno fino a detto secondo numero sequenziale di funzione decisionale in detta direzione data.

11. Un sistema od impianto a processore controllato da programma secondo la rivendicazione 9 o 10 comprendente inoltre:

il mezzo per memorizzare detto numero sequenziale del sistema come un numero sequenziale di processo per una pluralità di processi includenti tutti i processi che fanno richiami di funzioni per detta funzione di programma scelto al momento in cui è iniziata detta pluralità di processi;

mezzo per fare periodicamente secondi confronti tra i numeri sequenziali di processo di almeno tutti i processi attivi inclusi in detta pluralità, e detto numero sequenziale di funzione decisionale;

e mezzo per annullare detta versione corrente se nessuno di detti numeri sequenziali di processo di almeno tutti i processi attivi inclusi in detta pluralità è prima di

detto; numero sequenziale di funzione decisionale in detta data direzione.

12. In un sistema od impianto a processore controllato da programma, in cui sono iniziati processi di lavoro e sono eseguiti con richiami di funzioni a funzioni di programma memorizzate in una memoria di programma, mezzo per modificare una funzione di programma scelto durante l'esecuzione dei processi di lavoro, detto mezzo comprendente:

primo mezzo per memorizzare una versione corrente di detta funzione di programma scelto; secondo mezzo per memorizzare una versione modificata di detta funzione di programma scelto; mezzo per eseguire detta versione corrente/modificata se un processo che richiama detta funzione di programma scelto era iniziato prima che detta versione modificata fosse memorizzata in detto secondo mezzo, e

mezzo per eseguire detta versione modificata se detto processo che richiama detta funzione di programma scelto fosse iniziato dopo che detta versione modificata è stata memorizzata in detto secondo mezzo.

13. Un sistema od impianto a processore controllato da programma secondo la rivendicazione 12, in cui detto mezzo per eseguire detta versione corrente e detto mezzo per eseguire detta versione modificata comprendono inoltre:

il mezzo per mantenere un numero sequenziale del sistema

che è cambiato solo monotonicamente in una data direzione;

mezzo di controllo di processo per registrare detto numero sequenziale del sistema come un numero sequenziale di processo ogni volta che è iniziato un processo;

mezzo per generare e memorizzare un numero sequenziale di funzione decisionale avente un valore oltre il valore di qualsiasi numero sequenziale di processo in detta data direzione;

mezzo per cambiare detto numero sequenziale del sistema in un valore di almeno fino a detta direzione data come detto numero sequenziale di funzione decisionale;

mezzo rispondente ad un richiamo di funzione per detta funzione di programma scelto, per confrontare detto numero sequenziale di processo con detto numero sequenziale di funzione decisionale;

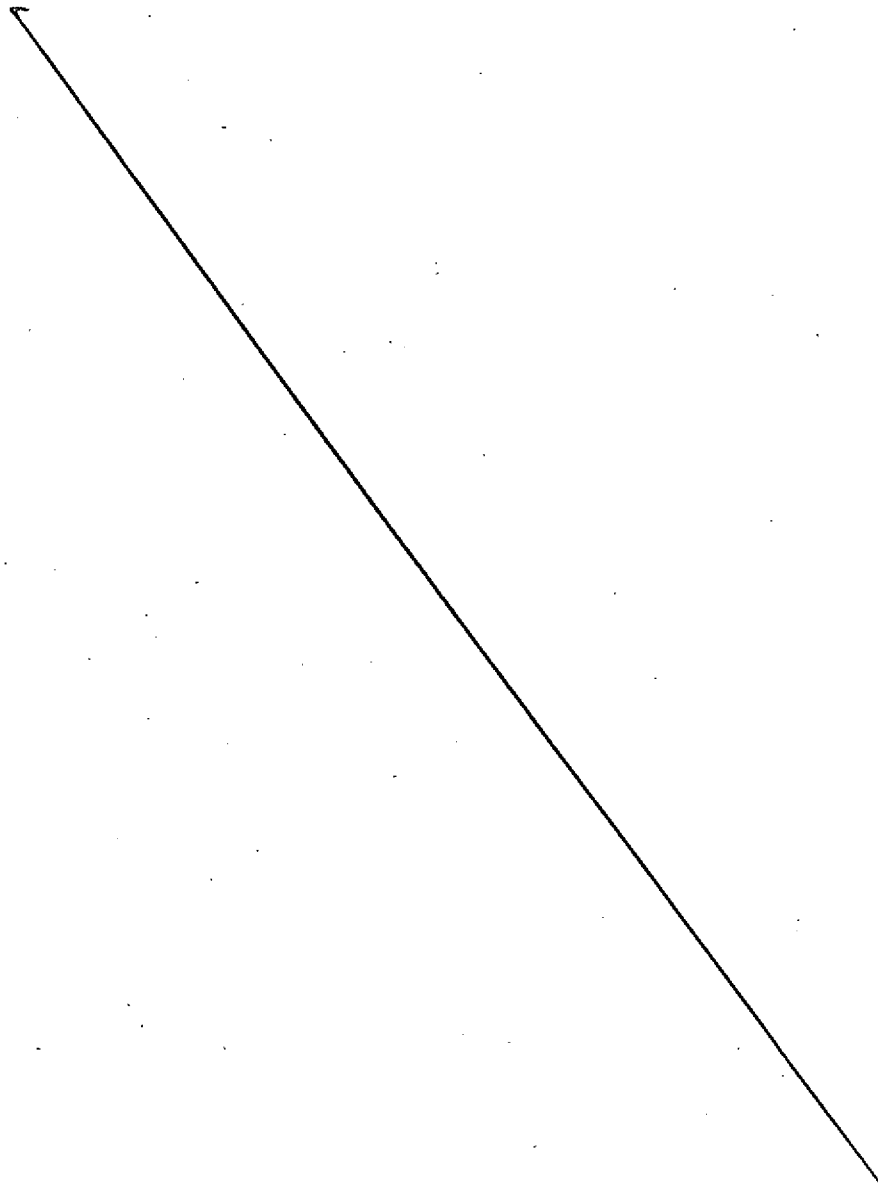
mezzo per eseguire detta versione corrente di detta funzione di programma se detto confronto indica che detto numero sequenziale di processo è ~~almeno fino a detto numero sequenziale di funzione decisionale in detta data direzione.~~

~~14. Un sistema od impianto a processore controllato da programma secondo la rivendicazione 13 in cui detto mezzo di controllo di processo comprende una posizione di memoria entro un blocco di controllo di processo associato~~

prima di detto numero sequenziale di funzione decisionale in detta data direzione; e

mezzi per eseguire detta versione modificata di detta funzione di programma se detta comparazione indica che detto numero sequenziale di processo è almeno fino a detto numero sequenziale di funzione decisionale in detta data direzione.

14. Un sistema od impianto a processore controllato da programma secondo la rivendicazione 13 in cui detto mezzo di controllo di processo comprende una posizione di memoria entro un blocco di controllo di processo associato



con detto processo, in cui detto numero sequenziale del sistema è memorizzato quando è iniziato detto processo.

15. In un sistema od impianto a processore controllato da programma in cui i processi di lavoro sono iniziati e sono; eseguiti con richiami di funzioni a funzioni di programma memorizzate in una memoria di programma,

mezzo per modificare una funzione di programma scelta durante l'esecuzione di processi di lavoro detto mezzo comprendente:

primo mezzo per memorizzare una versione corrente di detta funzione di programma scelta;

secondo mezzo per memorizzare una versione modificata di detta funzione di programma scelta;

terzo mezzo per memorizzare dati definenti un tempo di attivazione dopo il quale detta versione modificata di detta funzione di programma scelta deve essere impiegata; e

mezzo per eseguire detta versione corrente se un processo richiedente detta funzione di programma scelta è stato iniziato dopo detto tempo di attivazione.

16. Un sistema od impianto a processore controllato da programma secondo la rivendicazione 15 in cui detto mezzo per eseguire detta versione corrente e detto mezzo per eseguire detta versione modificata comprendono inoltre:

mezzo d'orologio del sistema per mantenere la data ed il tempo del sistema;

mezzo di controllo del processo per registrare detti data e tempo come un numero sequenziale di processo ogni volta che sia iniziato un processo;

mezzo per confrontare detto tempo di attivazione con detto numero sequenziale di processo prima dell'esecuzione di detta funzione di programma scelta;

mezzo per eseguire detta versione corrente se detto confronto indica che detto numero sequenziale di processo è prima di detto tempo di attivazione; e

mezzo per eseguire detta versione modificata di detta funzione di programma se detto confronto non indica che detto numero sequenziale di processo è prima di detto tempo di attivazione.

17. Un sistema od impianto a processore controllato da programma secondo la rivendicazione 16 in cui detto mezzo di controllo di processo comprende una posizione di memoria entro un blocco di controllo di processo associato con detto processo in cui detto tempo e data è memorizzato quando è iniziato detto processo.

ADT

FIG. 1

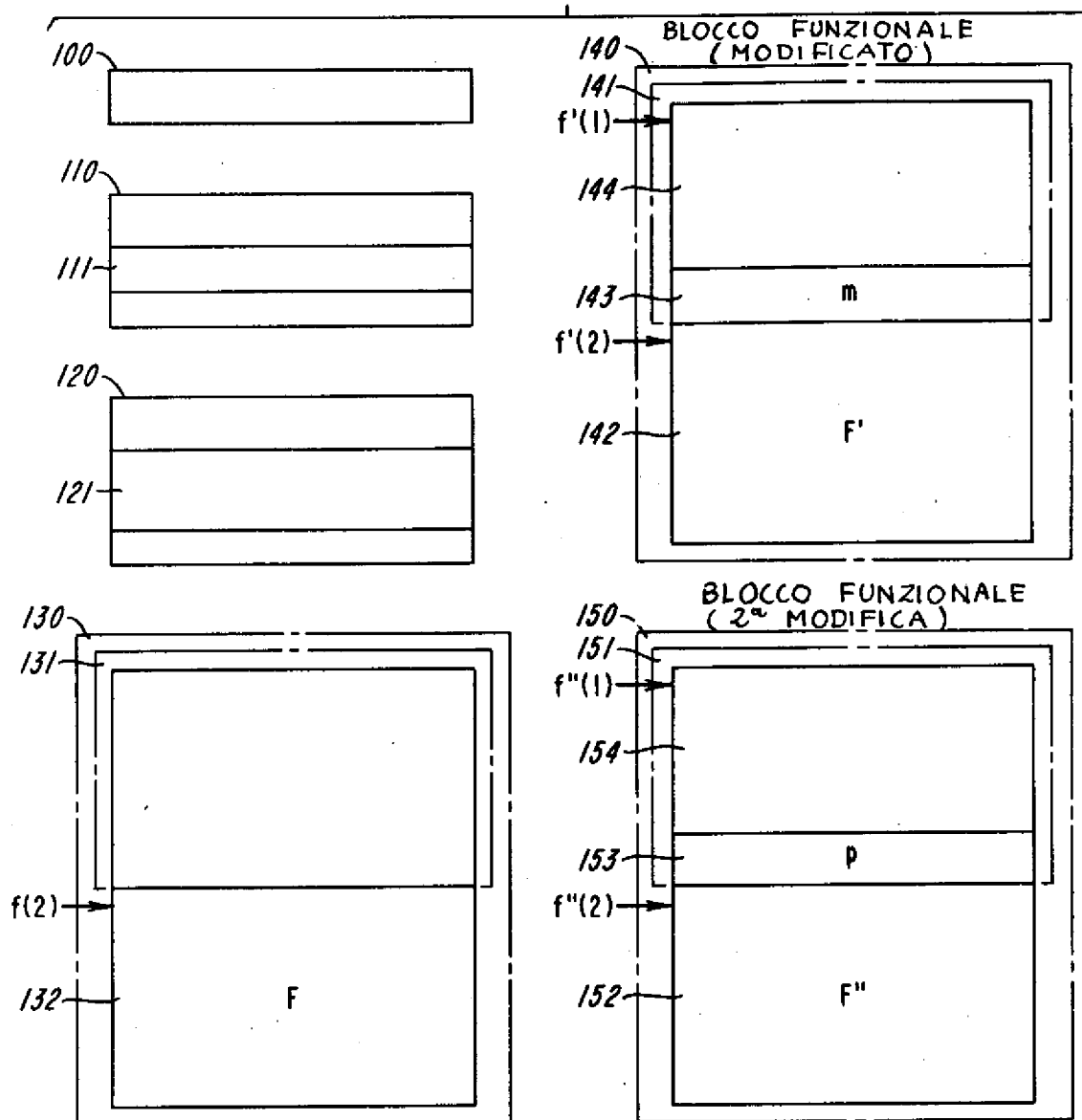
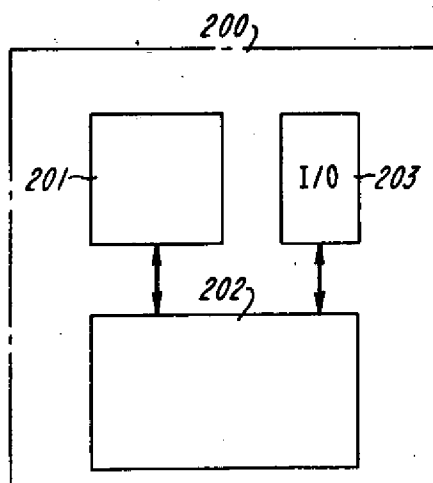


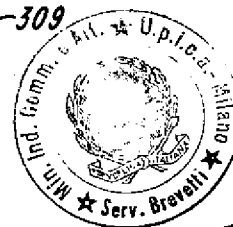
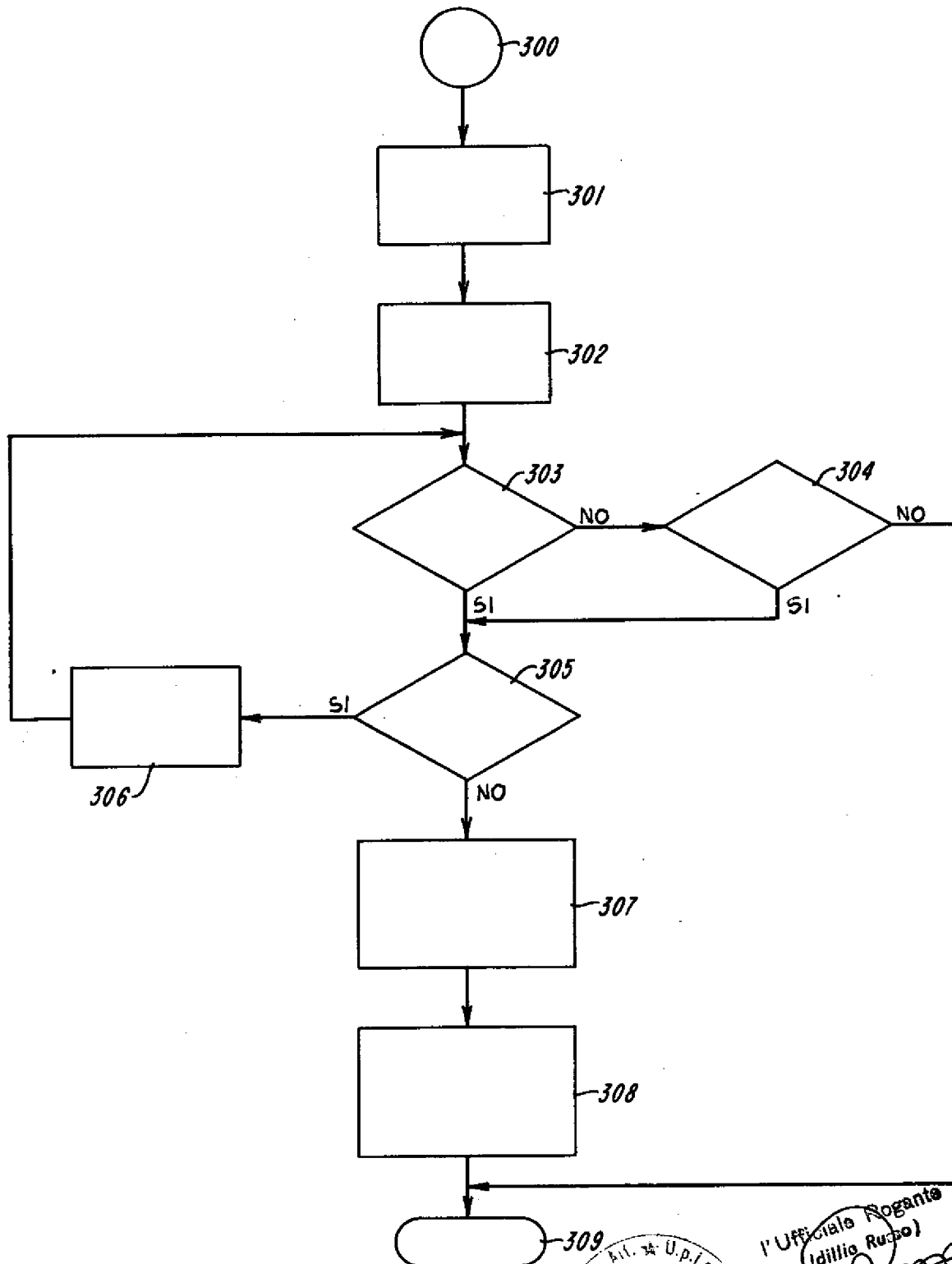
FIG. 2



l'Ufficio Rogante  
(Gilia Russo)

2/2

**FIG. 3**  
PROGRAMMA AUDIT



l'Ufficiale Rogante  
(Idillio Russo)

*[Handwritten signature]*

24362A/82

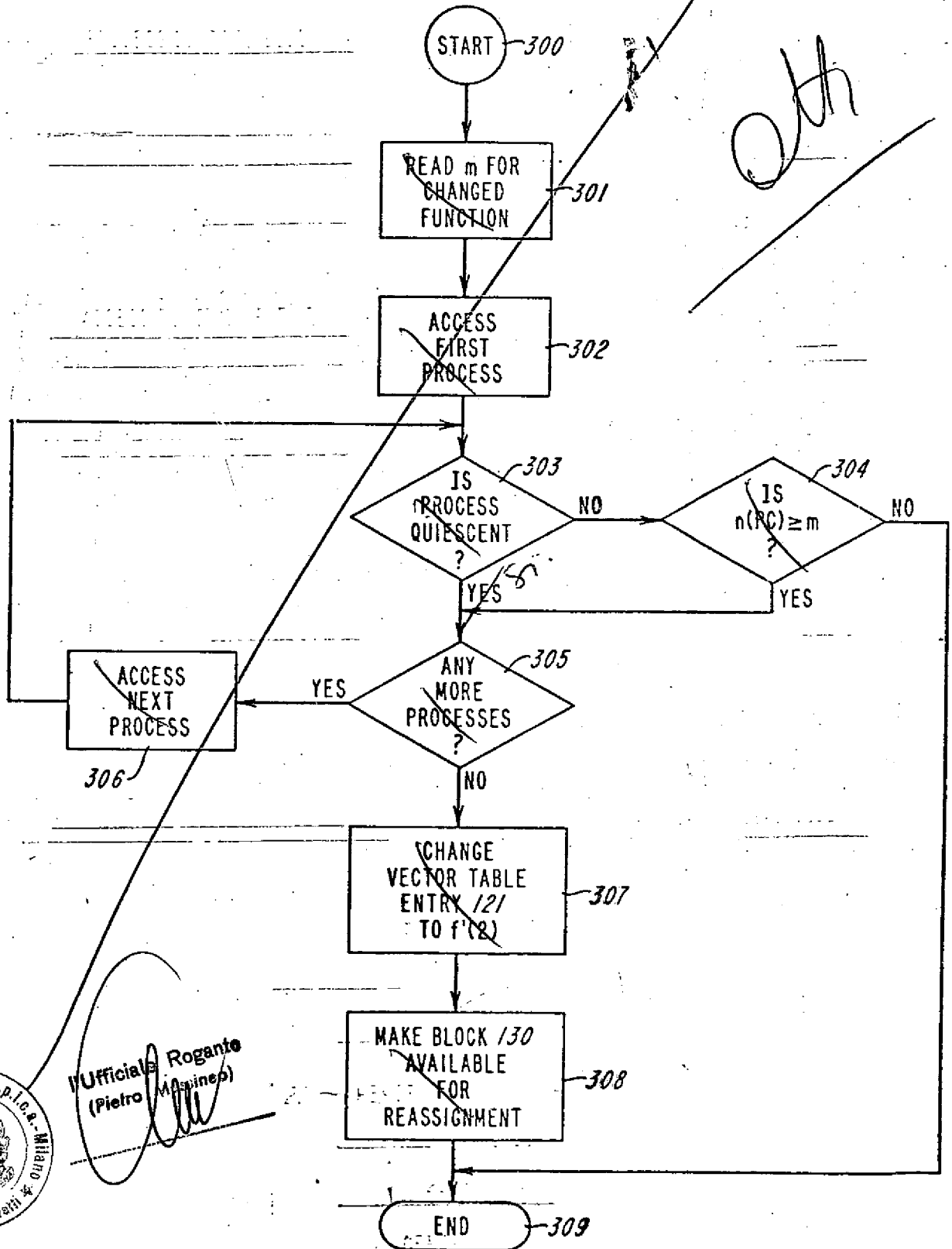
2/2

TAU II

FIG. 3  
AUDIT PROGRAM

*Programma audit*

*oth*



24362A/82

1/2

FIG. 1

*Block functional (modified) TAVI*

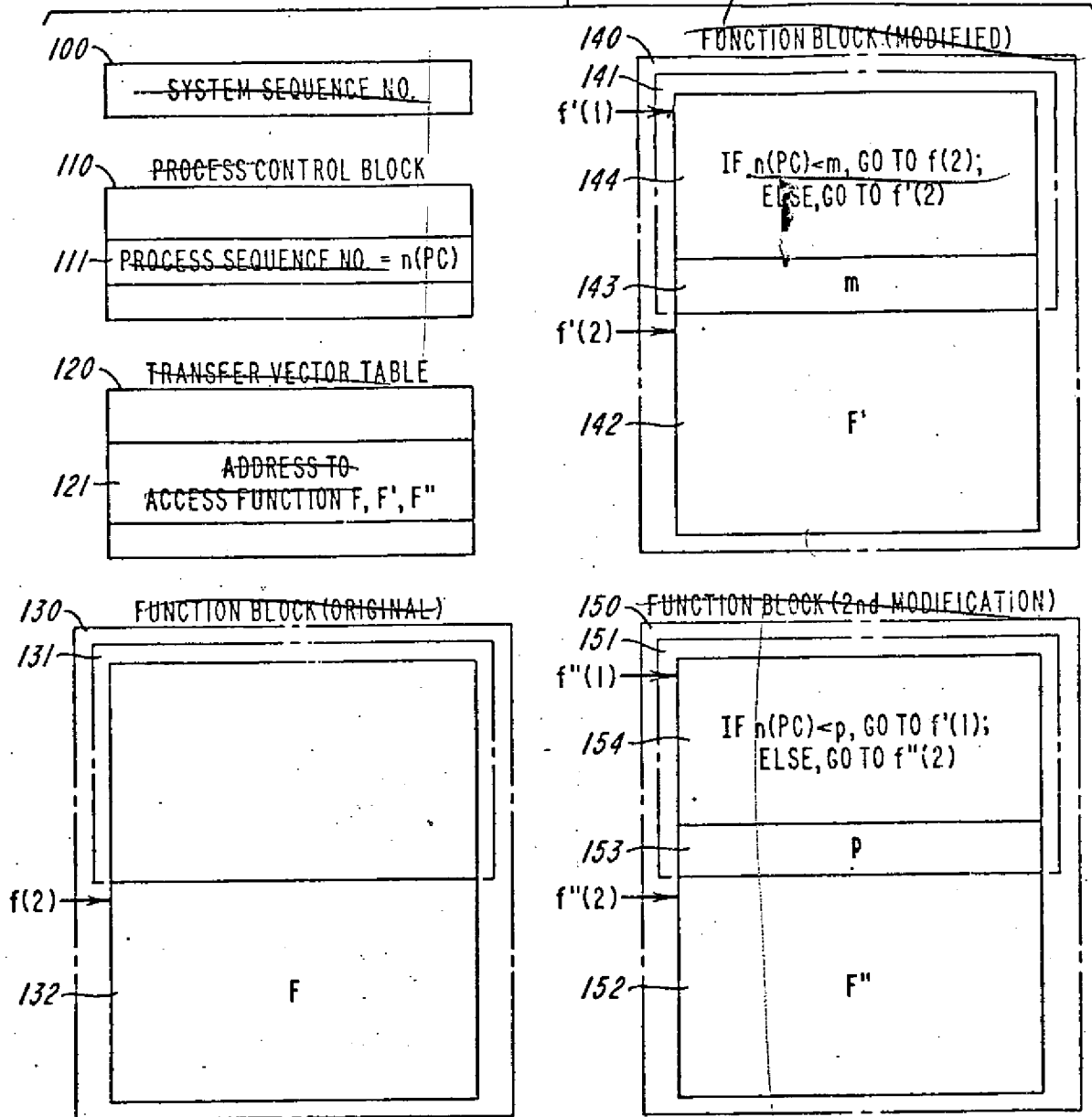
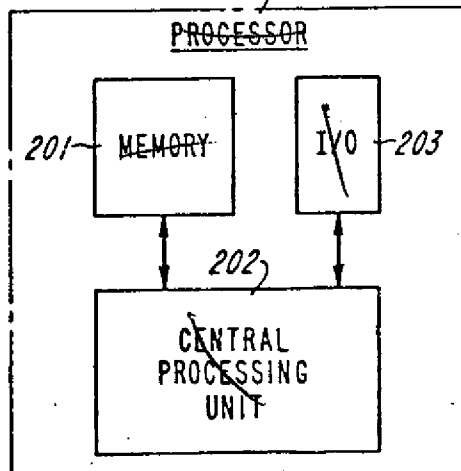
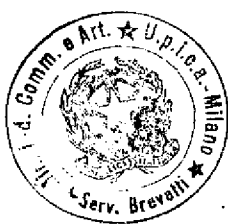


FIG. 2

200



*Block functional (2a modification)*



Ufficio Rogante  
(Pietro Mazzino)