

Consiglio Nazionale delle Ricerche

**Procedure per l'addebito e la fatturazione
delle risorse utilizzate dagli Utenti
del Servizio Elaborazione Dati**

P. Lazzereschi - M. Mannocci

154

CNUCE

A cura di: Pasquale Lazzereschi
Massimo Mannocci

Copyright - Novembre 1978

by - CNUCE - Pisa

Istituto del Consiglio Nazionale delle Ricerche

Procedure per l'addebito e la fatturazione delle risorse
utilizzate dagli utenti del Servizio Elaborazione Dati

F. LAZZARISCHI M. MANNOCCI

PREMESSA

La struttura centrale del servizio elaborazioni dati del C.N.U.C.E. e' costituita da due elaboratori IBM 370/168 e IBM 370/158 collegati tra loro da una linea ad alta velocita', in modo da consentire un interscambio di informazioni dati e programmi da l'uno a l'altro. I sistemi operativi sono rispettivamente per i due elaboratori:

370/158:	OS/VS2-HASP	Sistema a memoria virtuale per le elaborazioni in BATCH.
370/168:	VM-CMS-APL	Sistema a memoria
	OS/VS2-HASP	virtuale per le elaborazioni
	CICS-STAIRS	interattive ed in BATCH
	RSCS	Sistema multitasking per la gestione di linee B.S.C.

I sistemi operativi non sono rigidamente legati alla macchina, possono cioe' lavorare indistintamente sul 370/158 come sul 370/168.

Il sistema CICS-STAIRS a tuttoggi non produce dati di accounting.

Andiamo ora ad analizzare i dati prodotti da i vari sistemi.

Per quello che concerne l'OS/VS2-HASP la principale fonte dalla quale estrarre i dati di addebito, sono i records delle S.M.F. (System Management Facilitis), registrati sui data sets SYIS1.MANY e SYS1.MANY, sui quali vengono periodicamente memorizzate dall'OS tutte le informazioni concernenti i sistemi e i JOBS elaborati. Ai records S.M.F. ha accesso sia il sistema OS/VS2 sia il sistema HASP.

HASP scrive alcuni records con varie informazioni, come ad esempio: codice, nome del programmatore, nome del JOB, priorita', schede lette, linee scritte, elapsed time etc.

OS/VS2 scrive circa 30 tipi di records. Di questi alcuni presentano informazioni riguardanti il sistema (ora dell'IPL, unita' on-line, unita' off-line etc.) altri gli specifici JOBS (codice, programmatore, nome JOB, classe, memoria usata, cpu time, etc..

Il problema che si presenta a questo punto e' quello di estrarre dalla molteplicita' di informazioni della S.M.F. solamente quelle attinenti l'accounting. Chi provvede a cio' sono la serie di programmi, descritti nella pubblicazione CNUCE N. 16 "PROCEDURE PER LA RILEVAZIONE DEI DATI DI PRODUZIONE SOTTO OS/VS2" del dr. R. Bandinelli. La gestione standard dell'accounting del VM crea due soli tipi di records che contengono informazioni relative all'utilizzo:

- 1) di una macchina virtuale.
- 2) di un device dedicato (disco, nastro, etc.)

Vengono inoltre creati, in modo non standard, altri records che contengono informazioni sul numero di schede perforate, sulle pagine stampate, sui devices dedicati (disco, nastro, etc.)

Il limite piu' evidente della organizzazione standard e' l'uso delle schede come supporto esterno delle informazioni di accounting. E' stata percio' apportata una modifica affinche' le informazioni vengano raccolte direttamente su un disco che, una volta pieno, viene scaricato su nastro.

Il sistema ISCS crea solamente un record di addebito che contiene informazioni relative alla data e ora di inizio e fine delle attivita' del remoto, facendo successivamente gestire la registrazione dal VM.

Altri records di addebito sono creati dal programma di controllo descritto sulla pubblicazione CNUCE N. 66 "CONTROLLO E ADDEBITO AUTOMATICO DEI DATA SETS SU DISCO" del Dr. R. bandinelli, contenenti dati relativi alle dimensioni ed al periodo di vita dei data sets e dal programma RIS descritto dalla pubblicazione CNUCE N. 153 "PROCEDURE AUTOMATICHE PER LA GESTIONE DELLE RISORSE DI CALCOLO DEL SERVIZIO ELABORAZIONE DATI" del sig. Lazzareschi e Mannocci. Infatti il programma RIS oltre a gestire vari tipi di risorse, quali, archiviazione di nastri, dischi, terminali, vendite di materiale etc. , per ognuna di queste ne crea i records di addebito e li registra sull'archivio di nome FILE VARIAZ.

Rimangono quindi da analizzare i records di addebito dell'occupazione disco da parte di macchine virtuali. I programmi a cio' addetti sono MDISKMAP MDISKCK e AD1, descritti dalla pubblicazione CNUCE N. 153 DEL SERVIZIO ELABORAZIONE DATI. Essi registrano, al loro passaggio, la situazione nuova e scrivono la variazione rispetto alla

situazione precedente su un archivio di nome USER VARIAZ.

La raccolta dei dati di accounting non creati direttamente dall'elaboratore, viene gestita dalla procedura exec ADDEB, che lanciando vari programmi, va a scegliere tra i vari files di variazione FILE VARIAZ USER VARIAZ i records che interessano il periodo di addebito da prendere in esame.

E' conclusa cosi' la parte di registrazione dei records di addebito di tutte le risorse di calcolo fornite dal CNUCE.

Facendo ora un'analisi di quest'ultimi possiamo notare come essi, provenendo da molteplici fonti, siano molto diversi nei formati; si e' reso necessario percio' uniformarli. Il programma ADD01, seguendo una tabella descrittiva dei vari records, riesce a controllare la validita' e ad aggiungere davanti ad ognuno un campo di 40 bytes, contenente informazioni tipo (codice data etc.) in un formato uguale per tutti, in modo da poter selezionare i records per codice di addebito.

I dati validi vengono registrati da ADD01 su nastro magnetico; esso viene successivamente dato in input ad un programma di sort, in modo tale da ottenere un archivio ordinato sul codice di addebito dell'ente.

L'ultima fase e' quella vera e propria del conteggio e della stampa delle fatture. Essa viene eseguita dal programma ADD02 seguendo anch'esso una tabella che contiene e il formato dei vari records e i prezzi unitari delle risorse di calcolo. I dati in uscita saranno e la stampa su carta e la registrazione su disco delle varie fatture emesse.

Quest'ultime verranno usate per eseguire riespiloghi e statistiche di utilizzo trimestrali ed annuali. Le esecuzioni dei programmi di addebito ADD01 ed ADD02 sono completamente controllate da tabelle che rendono parametrici questi due programmi. E' quindi sufficiente modificare le tabelle senza toccare i programmi nel caso di cambiamento delle voci di addebito o di prezzo. L'unica limitazione e' che i dati di ingrasso devono avere formato scheda con un tipo scheda in posizione 79 80 e i dati riguardanti una stessa voce di addebito devono trovarsi tutti sullo stesso tipo scheda.

Capitolo I

Addebito risorse utilizzate

PROGRAMMA: SITNTD
LINGUAGGIO: PL1

ESTRAE DALLA SITUAZIONE NASTRI DISCHI E TERMINALI, RECORDS
PER L'ADDEBITO DI UN DETERMINATO PERIODO

FILES

SITUAZ NASTRI INPUT
SITUAZ DISCHI INPUT
SITUAZ TERMINAL INPUT
FILE VARIAZ INPUT
DATI ADD OUTPUT

Il programma legge a terminale la data di inizio e la data di fine del periodo di addebito nel formato GGMMAA. Legge i tre files di situazione, nastri dischi e terminali cercando i supporti riservati prima o entro il periodo di addebito; per ciascun supporto crea un record sul file DATI ADD registrandovi il numero di giorni di riserva del supporto all'interno del periodo di addebito. Legge il file variazioni ricercando tutte le registrazioni successive o interne al periodo di addebito e per ciascuna crea un record nel file DATI ADD nello stesso modo che per i files situazione.

FORMATO RECORD

FILE SITUAZ NASTRI, SITUAZ DISCHI, SITUAZ TERMINAL e FILE VARIAZ hanno il formato descritto nel programma sportello utenti

FILE DATI ADD TIPO SK 70

1-4	SIGLA RISERVA	CARATTERI
5-12	CODICE ADDEBITO	"
13-18	DATA RISERVA GGMMAA	"
19-20	TIPO SUPPORTO	"
21-49	NOME NASTRO DISCO O TERMINALE	"
50-57	PRIMI 8 CARATTERI NOME UTENTE	"
58-63	DATA INIZIO PERIODO DI ADDEBITO GGMMAA	"
64-69	DATA FINE PERIODO DI ADDEBITO GGMMAA	"
70-72	NUMERO DI GIORNI DA ADDEBITARE	"
73-78	DATA CANCELLAZIONE RISERVA GGMMAA (EVENTUALE)	"
79-80	TIPO SCHEDA 70	"

PROGRAMMA: ALDMV
LINGUAGGIO: PL1

Crea RECORDS per l'addebito dello spazio disco di macchine virtuali

FILE
USER VARIAZ INPUT (SORT SU NOME MV CODICE E DATA)
DATI ADD OUTPUT

Legge da terminale la data di inizio e la data di fine del periodo di addebito nel formato GGMMAA.
Per ogni M.V. nata prima o all'interno del periodo di addebito ADDMV prende in considerazione la prima variazione di configurazione; se creata prima del periodo di addebito viene presa in considerazione la data di inizio del periodo di addebito o altrimenti la data di creazione. Si calcola il numero di giorni fino alla prossima variazione all'interno del periodo di addebito, o sino alla fine del periodo di addebito se la M.V. non ha avuto variazioni e si scrive un record sul file DATI ADD.
Se la M.V. ha avuto piu' di una variazione durante il periodo di addebito viene scritto un record per ogni variazione.

FORMATO RECORD

FILE USER VARIAZ
FORMATO COME DESCRITTO NEL PROGRAMMA AD1
Sul file USER VARIAZ deve essere effettuato un sort sui campi NOME M.V. CODICE e DATA

FILE DATI ADD TIPO SK 71

1-8	NOME M.V.	CARATTERI
9-16	CODICE DI ADDEBITO	00
17-22	DATA VALIAZIONE	00
23-24	NUMERO DI CILINDRI DI 2314 MONTABILI	BINARIO
25-26	NUMERO DI CILINDRI DI 3330 MONTABILI	00
27-28	NUMERO DI CILINDRI DI 3340 MONTABILI	00
29-30	NUMERO DI CILINDRI DI 3350 MONTABILI	00
31-32	LIBERO	
33-36	NUMERO DI CILINDRI DI 2314	CARATTERI
37-40	NUMERO DI CILINDRI DI 3330	00
41-44	NUMERO DI CILINDRI DI 3340	00
45-48	NUMERO DI CILINDRI DI 3350	00
49-52	NUMERO GIORNI DA ADDEBITARE	00
53-58	DATA DI NASCITA M.V.	00
59-64	DATA CANCELLAZIONE M.V. (EVENTUALE)	00
65-66	LIBERO	00
67-72	DATA INIZIO PERIODO DI ADDEBITO	00
73-78	DATA FINE PERIODO DI ADDEBITO	00
79-80	TIPO SCHEDA 71	00

PROGRAMMA: AID01
LINGUAGGIO: PL1

Controlla la validita' delle schede addebito e le prepara per elaborazioni successive

FILE

SKINP4 INPUT (SK ADDEBITO NASTRI DISCHI TERMINALI M. V.CREATE DAI PROGRAMMI SITNTD e ADDMV)
SKINP3 INPUT (SK ADDEBITO DATA SET SYS1.ADDFILE)
SKINP1 INPUT (SK ADDEBITO JOB VS USCITA REPORT nastro W071 (GENNAIO-GIUGNO) W072 (LUGLIO-DICEMBRE))
SKINP2 INPUT (SK ADDEBITO VM nastro X071)
ANAG INPUT (FILE UTENTI DATASET ANAGRAF)
TABELLE INPUT (controlli di numericita' e costruzione del file FLC1, DATA SET ADDTAB12)
ERROR OUTPUT (nastro F016)
FLO1 OUTPUT (nastro F015)
SYSPRINT OUTPUT
SYSIN INPUT
STAMPA OUTPUT

Il programma legge dal FILE TABELLE i records con codice tabella 01 02 03 ed i parametri dal file SYSIN che sono:

```
DATA1='q1m1a1',DATA2='q2m2a2',<CONDUP=| 'SI' |>,<NGIORNI=nn>,  
| 'NO' |  
<NOME(1)='COD1NOME1',...NOME(15)='COD15NOME15'>;
```

q1m1a1 data di inizio del periodo di addebito
q2m2a2 data di fine del periodo di addebito

CONDUP='SI' viene fatto un sort sui record di input per eliminare i records duplicati

CONDUP='NO' non viene fatto ne' il sort ne' il controllo su records duplicati; se il parametro non specificato viene assunto CONDUP='NO'

NGIORNI=nn nn rappresenta il numero di giorni precedenti all'inizio del periodo di addebito o successivi allo stesso periodo entro cui si prevede siano compresi i records di input. Eventuali records con data minore di inizio periodo di addebito meno NGIORNI o con data maggiore di fine periodo di addebito piu' NGIORNI saranno segnalati come errore; se il parametro e' omesso viene assunto NGIORNI=99.

NOME(x)='codice nome JOB'

"Codice" e' il codice di addebito di 4 caratteri e

"nome JOB" e' il nome JOB o M.V. di 8 caratteri.
I records del file FL01 che hanno il codice uguale a
"codice" ed il nome JOB o M.V. uguale a "nome JOB"
vengono anche stampati sul file STAMPA con a destra il
numero di sequenza del record sul file FL01.
Questo parametro puo' essere utile per individuare i
records di addebito appartenenti ad un determinato
JOB, dato che non e' possibile in genere risalire
dalla stampa di un JOB ai suoi records di addebito.

I files SKINP1, SKINP2, SKINP3, SKINP4 sono letti
sequenzialmente uno dopo l'altro. La lettura termina al
primo file di cui non viene trovata la corrispondente scheda
DD, anche se ve ne sono di successive.

Per ogni record letto ADD01 controlla se esiste un
corrispondente record di tabella di tipo 01. Esegue i
controlli di numericita' sui campi indicati in tabella e
stampa eventuali messaggi di errore. ADD01 cerca in un
record di tabella di tipo 02 per quel tipo scheda (se non
viene trovato si ha il messaggio TIPO SK SCONOSCIUTA). Se
necessario viene convertita la data e confrontata con il
periodo di addebito; il record viene scartato se la sua data
e':

< DATA1 & > DATA1-NGIORNI | <DATA2 & < DATA2+NGIORNI

se invece la data e'

<DATA1-NGIORNI | >DATA2+NGIORNI

si ha la stampa del messaggio di errore.

* DATA CON SCARTO MAGGIORE DI NGIORNI *

Vengono quindi fatti tutti gli spostamenti indicati sul
record di tabella.

Se esiste un record di tabella di tipo 03 per quel tipo
scheda ed il campo confronto del record di tabella e' uguale
al campo indicato sul record del file SKINPx viene fatto il
trasferimento indicato.

Il codice di addebito viene controllato con i codici del
file ANAG; se non esiste si ha la stampa di un messaggio di
errore.

Al verificarsi di errori il record letto viene scritto sul
file EPRORI (nastro A158).

I records validi sono scritti sul nastro F015. Per fine dati
il programma termina stampando il totale delle schede lette
suddiviso per tipo scheda, il totale delle schede scritte
sui nastri F015, F016.

CODICE DI RITORNO

0 NON CI SONO STATI ERRORI
1 ERRORI NUMERICITA'
2 TIPO SCHEDE SCONOSCIUTA
3 DATA INVALIDA
4 DATA CON SCARTO MAGGIORE DI NGIORNI
5 CODICE SCONOSCIUTO

Se non esistono schede tipo 09 00 il codice di ritorno e' uguale ai precedenti piu' 10.

FORMATO RECORD

FILE: DATI ADD LRECL=80
tipi scheda 70 e 71 come descritto nei programmi
ADDMV e SITNDT

FILE: SKINP3 (SYS1.ADDFILE) LRECL=80 (SK ADDEBITO DATA SET)

1-4 CODICE DI ADDEBITO
5-9 DATA DI ADDEBITO YYDD
10-14 DATA ADDEBITO PRECEDENTE YYDD
15-17 NUMERO DI CILINDRI
18-23 NUMERO DI CILINDRI PER GIORNI
24-29 VOLUME SU CUI RISIESTE IL DATA SET
30-69 NOME DATA SET
70-78 LIBERO
79-80 TIPO SCHEDE 90

FILE: SKINP1 (NASTRO WC71 o successivi) LRECL=80 (SK ADDEBITO JOB VS)

formato e tipi scheda come descritto nel programma REPORT

FILE: SKINP2 (NASTRO XC71 o successivi) LRECL=80 (SK ADDEBITO VM)

SK MACCHINE VIRTUALI

1-8	NOME M.V.	C
9-16	CODICE ADDEBITO	C
17-22	DATA ADDEBITO MMGGAA	C
23-28	ORA ADDEBITO HHMMSS	C
29-32	TEMPO DI ON SEC.	B
33-36	TEMPO CPU TOTALE MILLISEC.	B
37-40	TEMPO CPU VIRTUALE MILLISEC.	B
41-44	NUMERO PAGINE LETTE	B
45-48	NUMERO PAGINE SCRITTE	B
49-52	NUMERO SIO PER I/O NON SPOOL	B
53-56	NUMERO SK PERFORATE IN SPOOL	B
57-60	NUMERO RIGHE STAMPATE IN SPOOL	B
61-64	NUMERO SCHEDE LETTE IN SPOOL	B

65-66	MEMORIA USATA IN K	B
67-70	TEMPO CONNESSIONE IN SECONDI	B
71-72	INDIRIZZO TERMINALE	B
73-78	LIBERO	
79-80	TIPO SCHEDA 01	C

C=CARATTERI

B=BINARIO

SCHEDA UNITA' DEDICATE E T-DISK

1-8	NOME M.V.	C
9-16	CODICE ADDEBITO	C
17-22	DATA ADDEBITO MMGGAA	C
23-28	ORA ADDEBITO HHMMSS	C
29-32	TEMPO CONNESSIONI DELL'UNITA' IN SEC.	B
33	CLASSE DELL'UNITA'	B
34	TIPO DELL'UNITA'	B
35	MODELLO (SE ESISTE)	B
36	CARATTERISTICA (SE ESISTE)	B
37-38	NUMERO CILINDRI T-DISK SOLO TIPO SCHEDA 03	B
39-78	LIBERO	
79-80	TIPO SK 02 o 03 PER UNITA' DEDICATE O T-DISK	C

C=CARATTERI

B=BINARIO

SCHEDA PAGINE STAMPATE O SCHEDE PERFORATE

1-8	NOME M.V.	C
9-16	CODICE ADDEBITO	C
17-22	DATA ADDEBITO MMGGAA	C
23-28	ORA ADDEBITO HHMMSS	C
29	FLAG P=PERFORAZIONE S=STAMPA	C
30-31	INDIRIZZO DEVICE	B
32-35	NUMERO PAGINE STAMPATE	B
36-39	NUMERO RIGHE STAMPATE O SK PERFORATE	B
40-41	NUMERO IDENTIFICAZIONE DEL FILE	B
42-78	LIBERO	
79-80	TIPO SCHEDA 05	C

C=CARATTERI

B=BINARIO

SCHEDA IPL VM

1-3	CARATTERI IPL	C
4-16	BLANKS	
17-22	DATA IPL MMGGSS	C
23-38	ORA IPL HHMMSS	C

29-78 LIBERO
79-80 TIPO SCHEDA 00 C

SCHEDA REMOTI RSCS

1-8 NOME M.V. CHE CREA LA SCHEDA C
9-16 NOME REMOTO C
17-24 DATA DISATTIVAZIONE MMGGAA C
25-32 ORA DISATTIVAZIONE HHMMSS C
33-66 LIBERO
67-70 TEMPO DI CONNESSIONE B
71-72 INDIRIZZO DI LINEA B
73-78 LIBERO
79-80 TIPO SCHEDA 09 C

X FILE TABELLE

TABELLA TIPO 01 CONTROLLO NUMERICITA'
Serve per il controllo di validita' dei campi numerici

1-2 TIPO TABELLA 01 *in genere non si fa*
3-10 ZERI
11-12 TIPO SCHEDA
13-14 POSIZIONE SULLA SCHEDA DEL CAMPO CHE DEVE ESSERE NUMERICO
15-16 LUNGHEZZA DEL CAMPO
i tre campi precedenti sono ripetuti 11 volte
77-80 LIBERO

TABELLA TIPO 02 COSTRUZIONE CHIAVE DI SELEZIONE

1-2 TIPO TABELLA 02
3-4 TIPO SCHEDA
5-6 POSIZIONE DATA SULLA SCHEDA
7 TIPO DATA
8-9 POSIZIONE ORA SULLA SCHEDA
10 TIPO ORA
11-12 POSIZIONE SULLA SCHEDA DEL CAMPO DA TRASFERIRE
13-14 LUNGHEZZA
15-16 POSIZIONE IN CUI DEVE ESSERE TRASFERITO SUL FILE DI USCITA
i tre campi precedenti sono ripetuti 11 volte
77-80 LIBERO

TIPO	DATA=1	DATA	IN	FORMA	MMGGAA	CARATTERI
	2	"	"	"	GGMMAA	"
	3	"	"	"	GG/MM/AA	"
	4	"	"	"	AAAGGG	"
	5	"	"	"	AAAGGG	PACKED

TIPO	ORA=1	DATA	IN	FORMA	HHMMSS	CARATTERI
	2	"	"	"	HH:MM:SS	"

SORT ADDEBITO

FILE
INPUT NASTRO A013
OUTPUT UN NASTRO PER MESE
ANNI PARI DA Q225 a Q236
ANNI DISPARI DA C201 a C212

Il sort viene fatto sui campi

1-4 119 9-20 33-40 120

Capitolo II

Riepiloghi mensili e stampa note di addebito

PROGRAMMA: ADD02
LINGUAGGIO: PL1

CALCOLO ADDEBITO E STAMPA RIEPILOGHI MENSILE DETTAGLIO
RISORSE UTILIZZATE E NOTE DI ADDEBITO

FILE

FILE TABELLE	INPUT
FILE UTENTI	INPUT
FLO1	INPUT NASTRO USCITA SORT F0xx o F0yy
PROSP1	OUTPUT (STAMPA CODICI INTERNI CRUCE)
PROSP2	OUTPUT (STAMPA CODICI TESI ED ESERCITAZIONI)
PROSP3	OUTPUT (STAMPA CODICI PAGANTI)
PROSP4	STAMPA NOTE DI ADDEBITO (CODICI NON PAGANTI)
PROSP5	STAMPA NOTE DI ADDEBITO (CODICI PAGANTI)
TOT	OUTPUT (STAMPA TOTALI)
meSE FATxx	OUTPUT (FILE DI ADDEBITO)
SYSIN	INPUT (EVENTUALI PARAMETRI)
SYSPRINT	OUTPUT (MESSAGGI DI ERRORE)
COD	OUTPUT (CODICI CON IMPORTO INFERIORE A 10000 lit.)

Inizialmente vengono lette le tabelle di tipo 04 05 06 e vengono chiesti a terminale eventuali parametri, questi possono essere:

```
      |'S'|      |'S'|  
<ADD=|      |>,<RISORSE=|'T'|>,<CODICI='COD'>;  
      |'N'|      |'N'|
```

ADD='S' viene fatto il calcolo dell'addebito e la stampa del riepilogo mensile.

ADD='N' non viene calcolato l'addebito.

RISORSE='S' viene fatta la stampa del dettaglio di tutti i jobs eseguiti e delle sessioni di M.V.

RISORSE='T' come per RISORSE='S', in piu' si ha la stampa dei supporti di calcolo in effetto (NASTRI, DISCHI, MACCHINE VIRTUALI, DATA SETS).

RISORSE='N' non viene fatta alcuna stampa di dettaglio.

CODICI='COD' COD e' una sequenza di codici senza caratteri separatori con un massimo di 20 se specificato viene fatto l'addebito solo di questi codici.

Se non non ci sono parametri dare un ritorno carrello.

ELABORAZIONE

Inizialmente legge i file "mese FATxx" di tutti i mesi precedenti, dello stesso anno per controllare e scrivere a terminale l'ultimo numero di nota di addebito emessa tanto per i codici paganti che per i codici non paganti. Se il numero di nota di addebito evidenziato e' corretto rispondere, alla successiva domanda, con un ritorno carrello altrimenti specificare il numero corretto. Per ogni record letto dal nastro viene cercato un record di tabella di tipo 04 con lo stesso tipo scheda del record letto. Se viene trovata ed il campo confronto della tabella e' uguale al campo indicato, sul record+40 bytes avvengono i trasferimenti indicati, altrimenti si passa subito al calcolo dell'addebito.

Letto un record si cerca il suo tipo scheda nei records di tabella di tipo 05. In caso di tipo scheda non trovato viene stampato un messaggio di errore ed il programma termina. Al contrario in caso di presenza del tipo scheda nella tabella 05, si analizza il campo "CONFRONTO" della stessa. Se tale campo risulta uguale a blank e' possibile calcolare l'addebito con il record individuato in tabella, altrimenti in caso di non uguaglianza a blank si analizzano i campi "POS" e "LUNG" del record di tabella e si usano come puntatore e lunghezza per individuare i dati dal record letto da nastro (file PL01); tali dati vengono ulteriormente confrontati con il campo "CONFRONTO" di tabella.

In caso di uguaglianza e' possibile calcolare l'addebito con il record individuato in tabella, altrimenti si ripetono le operazioni precedenti sino a quando non verra' rispettata una delle due condizioni di uguaglianza.

I vari campi del record letto, indicati dalla tabella in POSSK+40 vengono moltiplicati tra loro (dopo le necessarie conversioni) e per il fattore moltiplicativo indicato sulla tabella stessa.

Il risultato viene sommato all'elemento di indice (CODICE VOCI DI ADDEBITO) di un vettore MAT che contiene per ogni codice le risorse utilizzate.

Il record viene stampato in dettaglio su uno dei tre files PROSP1, PROSP2 e PROSP3 a seconda del suo codice tramite la subroutine STAMPA.

Per cambio del codice di addebito in input viene stampata la nota di addebito del codice precedente nel seguente modo.

Vengono esaminati gli elementi del vettore MAT e per quelli diversi da zero si ricerca una tabella di tipo 06, con il campo CODICE VOCI DI ADDEBITO uguale all'indice del vettore ed il tipo tariffa uguale al tipo tariffa indicato sul file

anagrafico per quel codice di addebito (se non viene trovato si ha la stampa di messaggio di errore ed il programma termina).

Una volta trovato si stampa DESCRIZIONE ADDEBITO e PREZZO BASE contenuti nella tabella, la quantita' contenuta nel vettore MAT e l'importo parziale ottenuto moltiplicando la quantita' per il prezzo base.

Dopo che si sono esauriti tutti gli elementi del vettore MAT viene stampata la somma degli importi parziali che da' l'importo totale.

Le stesse informazioni del riepilogo mensile sono scritte anche sul file "mese FATxx".

Finita la stampa del riepilogo viene azzerato il vettore MAT e si ritorna al calcolo dell'addebito.

Per fine dati viene stampato il riepilogo generale del mese ed il riepilogo per ogni gruppo di codici sul file TOT.

FORMATO RECORD

FILE SOLDI LRECL=26

per ogni codice si ha un record

1-4	CODICE ADDEBITO	CARATTERI
5-14	IMPORTO NOTA DI ADDEBITO EMESSA	"
15-19	numero nota di addebito	"
20	SPAZIO	
21-26	DATA DI EMISSIONE NOTA DI ADDEBITO GGMMAA	"
21-26	DATA FINE PERIODO DI ADDEBITO GGMMAA	"
	seguito da tanti records del formato seguente quante sono le voci di addebito per quel codice	
1-2	CODICE VOCI DI ADDEBITO	CARATTEFI
3	TIPO TARIFFA	"
3-14	QUANTITA' UTILIZZATA	"
15-26	SPESA PARZIALE	"

FILE TABELLE LRECL=80

TABELLA MODIFICA CAMPI TIPO 04

1-2	CODICE TABELLA 04
3-4	TIPO SCHEDA
5-10	ZERI
11-12	POSIZIONE SULLA SCHEDA DEL CAMPO DA CONFRONTARE
13-14	LUNGHEZZA
15-34	CAMPO CONFRONTO
35-36	POSIZIONE SULLA SK DOVE TRASFERIRE IL "CAMPO DA TRASFERIRE"
37-38	CAMPO DA TRASFERIRE
59-80	LIBERO

TABELLA CALCOLO ADDEBITO TIPO 05

1-2 CODICE TABELLA 05
3-4 TIPO SCHEDA
5-10 ZERI
11-12 CODICE VOCI DI ADDEBITO
13-32 FATTORE MOLTIPLICATIVO
33-34 POSIZIONE SULLA SCHEDA (CAMPO DA MOLTIPLICARE)
35-36 LUNGHEZZA
37 TIPO DATO 1=BINARIO 2=CARATTERI 3=PAKED
i tre campi precedenti sono ripetuti 7 volte
68-69 POSIZIONE SULLA SCHEDA DEL CAMPO DA CONFRONTARE
70-71 LUNGHEZZA
72-80 CAMPO CONFRONTO

TABELLA DESCRIZIONE ADDEBITO TIPO 06

1-2 CODICE TABELLA 06
3-4 CODICE VOCI DI ADDEBITO
5-19 PREZZO BASE
20 TIPO TARIFFA
21-69 DESCRIZIONE ADDEBITO
70 BLANK
71-80 UNITA' DI MISURA

PROGRAMMA: SUCC
LINGUAGGIO: PL1

RECUPERA I RECORD DI ADDEBITO PER I CODICI CHE UN IMPORTO INFERIORE A 10000 LIRI

FILE

FL01 INPUT (Stesso nastro usato in input da add02)
COD INPUT (File "COD COD creato da ADD02)
COD1 INPUT (Record di addebito corrispondenti ai codici presenti nel
file "COD COD")

Legge il file FL01 e trascrive i records i cui codici sono
presenti nel file "cod cod sul file "COD1 COD1" cambiando la
data del mese in quella del mese successivo

PROGRAMMA: DIVREC
LINGUAGGIO: PL1

Divide i record del file "codici M10000" in modo da poterli
trasferire tramite punch al sistema os.

PROGRAMMA: SOMREC
LINGUAGGIO: PL1

Ricomponi i record divisi precedentemente dal programma
DIVREC.

Capitolo III

Elenco delle Procedure

PROCEDURA EXEC ADDEB

Creazione file contenente le schede addebito di nastri dischi terminali e macchine virtuali per il mese di addebito richiesto (FILE DATI ADD).

Richiama la procedura MESE, fa il link del disco di ADMIN dove si trova il file USER VAEAZ e richiama la procedura ADDSK, fa il link alla M.V. UTENTI dove si trovavano i files situazione e variazione di nastri dischi e terminali, richiama la procedura ADDN.

```
&CONTROL OFF NOMSG
STATE DATI ADD A1
&IF &RETCODE EQ 0 &GOTO -ER1
EXEC MESE
&G1 = &GLOBAL1
&IF &GLOBAL1 LT 10 &G1 = &CONCAT 0 &GLOBAL1
&M1 = &GLOBAL2
&IF &GLOBAL2 LT 10 &M1 = &CONCAT 0 &GLOBAL2
&M2 = &M1
&A1 = &GLOBAL3
&A2 = &A1
&G2 = &GLOBAL4
* ACCESSO DISCO 201 CORRISPONDE AL 197 DI ADMIN
ACC 201 Z/A
EXEC ADDSK &G1 &M1 &A1 &G2 &M2 &A2
* ACCESSO DISCO 202 CORRISPONDE AL 191 DI UTENTI
REL Z
ACC 202 Z/A
* ACCESSO DISCO 203 CORRISPONDE AL 222 DI 99999999 SIT NASTR DISC
CP LINK 99999999 222 203 RR TAPESTAT
ACC 203 F/A
FI ADD DISK DATI ADD A1 (RECFM F BLOCK 80 DISP MOD
EXEC ADDN &G1 &M1 &A1 &G2 &M2 &A2
REL Z
REL F (DET
&EXIT
-ER1 &TYPE DATI ADD A1 ESISTE GIA'
```

PROCEDURA EXEC MESE

Viene richiamata da altre procedure, chiede a terminale MESE ed ANNO nella forma mese per esteso e anno ultime due cifre. La procedura calcola le date di inizio e di fine mese e le restituisce nelle variabili.

&GLOBAL1 &GLOBAL2 &GLOBAL3 &GLOBAL4 &GLOBAL5 &GLOBAL2
GG MM AA GG MM AA

```
&IF &INDEX EQ 2 &GOTO -SE
-A &TYPE MESE E ANNO
&READ ARGS
&IF &INDEX NE 2 &GOTO -ER1
-SE &IF &2 LT 61 &GOTO -ER1
&IF &2 GT 99 &GOTO -ER1
&M = 01
&GLOBAL4 = 31
&IF &1 EQ GENNAIO &GOTO -OK
&M = 02
&IF &1 EQ FEBBRAIO &GOTO -OK
&M = 03
&GLOBAL4 = 31
&IF &1 EQ MARZO &GOTO -OK
&M = 04
&GLOBAL4 = 30
&IF &1 EQ APRILE &GOTO -OK
&M = 05
&GLOBAL4 = 31
&IF &1 EQ MAGGIO &GOTO -OK
&M = 06
&GLOBAL4 = 30
&IF &1 EQ GIUGNO &GOTO -OK
&M = 07
&GLOBAL4 = 31
&IF &1 EQ LUGLIO &GOTO -OK
&M = 08
&GLOBAL4 = 31
&IF &1 EQ AGOSTO &GOTO -OK
&M = 09
&GLOBAL4 = 30
&IF &1 EQ SETTEMBRE &GOTO -OK
&M = 10
&GLOBAL4 = 31
&IF &1 EQ OTTOBRE &GOTO -OK
&M = 11
&GLOBAL4 = 30
&IF &1 EQ NOVEMBRE &GOTO -OK
&M = 12
```

```
&GLOBAL4 = 31
&IF &1 EQ DICEMBRE &GOTO -OK
-ER1 &TYPE ERRORE MESE O ANNO
&GOTO -A
-OK &IF &M NE 2 &GOTO -OK1
&C = 60
&LOOP 2 25
&C = &C + 4
&IF &C EQ &2 &GOTO -B
&GLOBAL4 = 28
&GOTO -OK1
-B &GLOBAL4 = 29
-OK1 &GLOBAL1 = 01
&GLOBAL2 = &M
&GLOBAL3 = &2
&GLOBAL5 = &M
&GLOBAL6 = &2
&EXIT
```

PROCEDURA EXEC NASTROAD

Vuole come parametri il mese e l'anno di addebito, il mese come numero da 1 a 12 e l'anno ultime due cifre. Calcola la sigla del nastro corrispondente anni pari da F001 a F012 anni dispari da F021 a F032 e mette questa sigla in STACK di input da terminale in ordine LIFO.

```
&D = 000
&P = 020
&X = 60
-Q &X = &X + 2
&IF &X EQ &2 &GOTO -PARI
&IF &X LT &2 &GOTO -Q
&N1 = &D + &1
&IF &N1 < 10 &N1 = &CONCAT 0 &N1
&NA = &CONCAT F 0 &N1
&GOTO -VAI
-PARI &N1 = &P + &1
&NA = &CONCAT F 0 &N1
-VAI &STACK LIFO &NA
```

PROCEDURA EXEC ADDSK

Fa un sort sul file USER VARIAZ e manda in esecuzione il programma ADDMV con i necessari FILEDEF.

```
&CONTROL OFF NOMSG
&DA1 = &CONCAT &1 &2 &3
&DA2 = &CONCAT &4 &5 &6
ERASE USER1 VAF
&STACK LIFO 1 8 21 22 17 18 19 20 23 28 9 16
SORTF USER VARIAZ Z USER1 VAR A1
FI VAR DISK USER1 VAR (RECFM F BLOCK 80
FI ADD DISK DATI ADD A1 (RECFM F BLOCK 80
&STACK LIFO &DA2
&STACK LIFO &DA1
ADDMV
ERASE USER1 VAF
```

PROCEDURA EXEC ADDN

Manda in esecuzione il programma SITNTD con i necessari FILEDEF

```
&CONTROL OFF NOMSG
&DAT1 = &CONCAT &1 &2 &3
&DAT2 = &CONCAT &4 &5 &6
FI SITUAZ DISK SITUAZ NASTRI F ( RECFM F BLOCK 80
FI VARIAZ DISK FILE VARIAZ Z ( RECFM F BLOCK 80
FI SITUAZD DISK SITUAZ DISCHI F (RECFM F BLOCK 80
FI SITUAZT DISK SITUAZ TERMINAL F (RECFM F BLOCK 80
&STACK LIFO &DAT2
&STACK LIFO &DAT1
SITNTD
```

PROCEDURA EXEC ADD01

Richiama inizialmente la procedura MESE e poi invia al VS/158 le schede controllo per l'esecuzione dei programmi ADD01 ed RSCS.

```
&TYPE MESE E ANNO
&READ VARS &MESE &AAN
EXEC MESE &MESE &AAN
&GGGG = &GLOBAL1
&IF &GLOBAL1 LT 10 &GGGG = &CONCAT 0 &GLOBAL1
&MMMM = &GLOBAL2
&IF &GLOBAL2 LT 10 &MMMM = &CONCAT 0 &GLOBAL2
&DAT1 = &CONCAT ' &GGGG &MMMM &GLOBAL3 '
&DAT2 = &CONCAT ' &GLOBAL4 &MMMM &GLOBAL3 '
&GOTO -OKK
-OK &DAT1 = &CONCAT ' &1 &2 &3 '
&DAT2 = &CONCAT ' &4 &5 &6 '
-OKK CP SPOOL PUN CONT RSCS1
CP TAG DEV D REMOS
&IF &GLOBAL2 GT 6 &GOTO -DOPO
&Z = 2SEM78
-COM &LL2 = &GLOBAL2 + &GLOBAL2
&LL1 = &LL2 - 1
&GOTO -VAI
-DOPO &GLOBAL2 = &GLOBAL2 - 6
&Z = 2SEM78
&GOTO -COM
-VAI &BEGPUNCH
//ADD01 JOB (0000,K210,20,8), 'ADDEBITO1'
/*SETUP          TPV9=1
/*MESSAGE        2SEM78=W076
//UU EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT2 DD UNIT=SYSDA,SPACE=(CYL,(1,1)),DSN=&&SKNDT,
//          DCB=(RECFM=F,BLKSIZE=80),DISP=(NEW,PASS)
//SYSUT1 DD *
&END
PUN DATI ADD B1 (NOH
&BEGPUNCH
//U1 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT2 DD UNIT=SYSDA,SPACE=(80,(3,1)),DISP=(NEW,PASS),
//          DSN=&&PARIN,DCB=(RECFM=F,BLKSIZE=80)
//SYSUT1 DD *
&END
&PUNCH DATA1= &DAT1 DATA2= &DAT2 NGIORNI= 400 CONDUP= 'SI';
&BEGPUNCH
```



```
//AD1 PROC D1=X,D2=Y,L1=1,L2=2,NA=XXXX
//UNO EXEC PGM=ADD01,REGION=1024K PASSWORD='P1P1P1P1'
//STEPLIB DD UNIT=SYSDA,VOL=SER=USER01,DSN=ADDSTAT,DISP=SHR
// DD DSN=SYS1.PLOLINK,DISP=SHR
//SYSLIB DD DSN=SYS1.PLFLIB,DISP=SHR
// DD DSN=SYS1.FLRLIB2,DISP=SHR
// DD DSN=SYS1.SSPPL1,DISP=SHR
//PLIDUMP DD DUMMY
//SYSPRINT DD SYSOUT=A
//SYSOUT DD SYSOUT=A
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(100),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(100),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(100),,CONTIG)
//STAMPA DD SYSOUT=A
//ERROR DD UNIT=SYSDA,DSN=&&ERRO,DCB=(RECFM=F,BLKSIZE=80),
// DISP=(NEW,PASS),SPACE=(CYL,(10),,CONTIG)
//SKINP1 DD UNIT=(TPV9,,DEFER),LABEL=(2,BLP),DISP=(OLD,KEEP),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600),VOL=SER=(X073)
//SKINP2 DD UNIT=APF=SKINP1,LABEL=(&L1,SL),DISP=(OLD,KEEP),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=2480),VOL=SER=&NA,DSN=&D1..PARTE1
// DD UNIT=APF=SKINP1,LABEL=(&L2,SL),DISP=(OLD,KEEP),VOL=SER=&NA,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=2480),DSN=&D2..PARTE2
//SKINP3 DD DSN=SYS1.ADDFILE,DISP=SHR
//SKINP4 DD UNIT=SYSDA,DSN=&&SKNDT,DISP=(OLD,DELETE)
//SKINP5 DD UNIT=APF=SKINP1,LABEL=(2,BLP),DISP=(OLD,KEEP),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=1600),VOL=SER=(X074)
//FLO1 DD UNIT=APF=SKINP1,LABEL=(,BLP),DISP=(NEW,PASS),DSN=FL,
// DCB=(RECFM=FB,LRECL=120,BLKSIZE=6000),VOL=SER=F015
//TABELLE DD UNIT=SYSDA,VOL=SER=USER01,DISP=SHR,DSN=ADDTAB12
//ANAGRAF DD UNIT=SYSDA,VOL=SEP=USEP01,DISP=SHR,DSN=ANAGRAF
//SYSIN DD UNIT=SYSDA,DSN=&&PARIN,DISP=(OLD,DELETE)
// PEND
// EXEC AD1,
&END
&STACK LIFO FILE
&STACK LIFO O? &Z
&STACK LIFO I // NA=
&STACK LIFO C / $/,/
&STACK LIFO O? &MESE $
&STACK LIFO I // D1=
&STACK LIFO C / $/,/
&STACK LIFO O? &MESE $
&STACK LIFO I // D2=
&STACK LIFO C / $/,/
&STACK LIFO O? &LL1 $
&STACK LIFO I // L1=
&STACK LIFO C / $/,/
&STACK LIFO O? &LL2 $
&STACK LIFO I // L2=
&STACK LIFO TABSET 1 6
```

```
SET INPUT ? 05
ERASE $PARM$ $DSS$ A1
E $PARM$ $DSS$ A1
SET INPUT ? 6F
PUN $PARM$ $DSS$ A1 (NOH
ERASE $PARM$ $DSS$ A1
&BEGPUNCH
//QUAT EXEC PGM=IEBGENER,COND=(100,LT)
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=SYSDA,DSN=&&ERRO,DISP=OLD,DCB=(RECFM=F,BLKSIZE=80)
//SYSUT2 DD UNIT=(TPV9,,DEFER),LABEL=(,BLP),DISP=(NEW,KEEP),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),VOL=SER=P016
&END
CP SPOOL PUN NOCONT CLOSE
```

PROCEDURA EXEC ADD01ER

Serve per la correzione dei records con errori scritti sul nastro F016.

Unica differenza da ADD01, chiede un solo nastro di input (F016) ed ha DISP=MOD sul nastro di uscita.

```
&IF &INDEX EQ 6 &GOTO -OK
EXEC MESE
&GGG = &GLOBAL1
&IF &GLOBAL1 LT 10 &GGG = &CONCAT 0 &GLOBAL1
&MMM = &GLOBAL2
&IF &GLOBAL2 LT 10 &MMM = &CONCAT 0 &GLOBAL2
&DAT1 = &CONCAT ' &GGG &MMM &GLOBAL3 '
&DAT2 = &CONCAT ' &GLOBAL4 &MMM &GLOBAL6 '
&GOTO -Z
-OK &DAT1 = &CONCAT ' &1 &2 &3 '
&DAT2 = &CONCAT ' &4 &5 &6 '
-Z CP SPOOL PUN CONT RSCS1
CP TAG DEV D REMOS
&BEGPUNCH
//ADD01 JOB (0000,K210,04,5), 'ADDEBITO1'
/*SETUP TPV9=1
//UNO EXEC PGM=ADD01,REGION=1024K PASSWORD='P1P1P1P1'
//STEPLIB DD UNIT=SYSDA,VOL=SER=USER01,DSN=ADDSTAT,DISP=SHR
// DD DSN=SYS1.PLOLINK,DISP=SHR
//SYSLIB DD DSN=SYS1.PLELIB,DISP=SHR
// DD DSN=SYS1.PLRLIB2,DISP=SHR
// DD DSN=SYS1.SSPPL1,DISP=SHR
//PLIDUMP DD DUMMY
//SYSPRINT DD SYSOUT=A
//SYSOUT DD SYSOUT=A
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(100),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(100),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(100),,CONTIG)
//STAMPA DD SYSOUT=A
//ERROR DD UNIT=SYSDA,DSN=&&ERRO,DCB=(RECFM=F,BLKSIZE=80),
// DISP=(NEW,PASS),SPACE=(CYL,(3,2))
//SKINP1 DD UNIT=(TPV9,,DEFER),LABEL=(,BLP),DISP=(OLD,KEEP),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),VOL=SER=F016
//FLO1 DD UNIT=APP=SKINP1,LABEL=(,BLP),DISP=(MOD,PASS),DSN=PL,
// DCB=(RECFM=FB,LRECL=120,BLKSIZE=6000),VOL=SER=F015
//TABELLE DD UNIT=SYSDA,VOL=SER=USER01,DISP=SHR,DSN=ADDTAB12
//ANAGRAF DD UNIT=SYSDA,VOL=SER=USER01,DISP=SHR,DSN=ANAGRAF
//SORTIN DD UNIT=SYSDA,DSN=&&INSRT,DISP=(NEW,PASS),
// DCB=(RECFM=F,BLKSIZE=120),SPACE=(CYL,(5,3))
//SYSIN DD *
&END
&PUNCH DATA1= &DAT1 DATA2= &DAT2 NGIORNI= 300 CONDUP= 'SI';
&BEGPUNCH
```

```
//DUE EXEC SORTD,COND=(10,LT,UNO),REGION=512K
//SORTIN DD UNIT=SYSDA,DSN=&&INSRT,DISP=OLD,
//   DCB=(RECFM=F,BLKSIZE=120)
//SORTOUT DD UNIT=SYSDA,DSN=&&OUTSRT,DISP=(NEW,PASS),
//   SPACE=(CYL,(5,3)),DCB=(RECFM=F,BLKSIZE=120)
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(10),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(10),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(10),,CONTIG)
//SYSIN DD *
  SORT FIELDS=(9,12,CH,A,119,2,CH,A)
/*
//TRF EXEC PGM=ISCS,REGION=512K,COND=(10,LT)
//STEPLIB DD UNIT=SYSDA,VOL=SER=USER01,DISP=SHR,DSN=ADDSTAT
//   DD DSN=SYS1.PLOLINK,DISP=SHR
//SYSLIB DD DSN=SYS1.PLELIB,DISP=SHR
//   DD DSN=SYS1.PLELIB2,DISP=SHR
//   DD DSN=SYS1.SSPPL1,DISP=SHR
//PLIDUMP DD DUMMY
//SYSPRINT DD SYSOUT=A
//INP DD UNIT=SYSDA,DSN=&&OUTSRT,DISP=OLD,DCB=(RECFM=F,BLKSIZE=120)
//OUT DD UNIT=TPV9,DISP=(MOD,KEEP),DSN=FL,
//   VOL=SER=FC15,LABEL=(,BLP),
//   DCB=(RECFM=FB,LRECL=120,BLKSIZE=6000)
//QUAT EXEC PGM=IEBGENER,COND=(100,LT)
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=SYSDA,DSN=&&ERRO,DISP=OLD,DCB=(RECFM=F,BLKSIZE=80)
//SYSUT2 DD UNIT=(TPV9,,DEFER),LABEL=(,BLP),DISP=(NEW,KEEP),
//   DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),VOL=SER=FO16
&END
CP SPOOL PUN NOCONT CLOSE
```

PROCEDURA EXEC SORTAD

SORT ADDEBITO

Richiama la procedura MESE, chiama la procedura NASTROAD per calcolare il numero del nastro di uscita, manda in esecuzione il programma DIVREC per dividere i record lunghi 120 del file "CODICI M10000" in modo da poterli trasferire tramite il punch, poi invia al vs/158 il programma SOMREC per riunire i record precedentemente divisi e le schede controllo per l'esecuzione del SORT dei dati dal file "CODICI M10000" e dal nastro F015 sul nastro con label calcolata per quel mese di addebito.

```

CP TAG DEV D REMOS
CP SPOOL PUN CONT RSCS1
EXEC MESE
&M = &GLOBAL2
&A = &GLOBAL3
EXEC NASTROAD &M &A
&READ VARS &NA
&BEGPUNCH
//SORTAD JOB (0000,K210,008,01),'$ACCT01 SORT AD',REGION=2000K
/*SETUP          TPV9=1
&END
&STACK LIFO FILE
&CCCC = &CONCAT / Z / &NA /
&STACK LIFO C &CCCC
&BEGSTACK LIFO
I /*MESSAGE          NASTFO Z CON ANELLO
&END
ERASE $$$$ $$$$ A1
EDIT $$$$ $$$$ A1
PUN $$$$ $$$$ A1 (NOH
ERASE $$$$ $$$$ A1
&BEGPUNCH
// EXEC PLO
//COMP.SYSIN DD *
SOMREC:PROC OPTIONS(MAIN);
DCL REC CHAR(120),P1 CHAR(80) DEF REC,P2 CHAR(40) DEF REC POS(81);
DCL INP FILE INPUT ,OUT FILE OUTPUT RECORD;
ON ENDFILE(INP) GOTO FINE;
LE: GET FILE(INP) EDIT(P1,P2) (COL(1),A(80),SKIP,A(40));
WRITE FILE(OUT) FROM(REC);
GOTO LE;
FINE:END;
//GO.OUT DD UNIT=SYSDA,DSN=&&CODM10,SPACE=(CYL,(2,1)),VOL=SER=WORK01,
// DCB=(RECFM=F,BLKSIZE=120,LE ECL=120),DISP=(NEW,PASS)
//GO.INP DD *
&END

```

```

SOM00010
SOM00020
SOM00030
SOM00040
SOM00050
SOM00060
SOM00070
SOM00080

```

```
FI INP DISK CODICI M10000 (RECFM F BLOCK 120
FI OUT PUNCH (RECFM F BLOCK 80
LOAD DIVREC (START
&BEGPUNCH
//SORTC PROC N=XXXX
// EXEC PGM=SORTAD          PASSWORD='P1P1P1P1'
//STEPLIB DD DSN=SYS1.SMLINK,DISP=SHR
//          DD UNIT=SYSDA,VOL=SER=USER01,DSN=ADDSTAT,DISP=SHR
//          DD DSN=SYS1.PLOLINK,DISP=SHR
//SORTLIB DD DSN=SYS1.SMLIB,DISP=SHR
//SYSLIB DD DSN=SYS1.PLFLIB,DISP=SHR
//          DD DSN=SYS1.PLRLIB2,DISP=SHR
//          DD DSN=SYS1.SSPPL1,DISP=SHR
//SYSOUT DD SYSOUT=A
//SORTIN  DD UNIT=TPV9,VOL=SER=F015,DSN=SMF,
//          DISP=(OLD,KEEP),LABEL=(,BLP),
//          DCB=(RECFM=FB,LRECL=120,BLKSIZE=6000)
//          DD UNIT=SYSDA,DSN=&&CODM10,VOL=SER=WORK01,DISP=OLD
//OUT DD UNIT=&FF=SORTIN,VOL=SER=&N,LABEL=(,BLP),
//          DISP=(NEW,KEEP),
//          DCB=(RECFM=FB,LRECL=120,BLKSIZE=6000,DEN=3)
//GIAN DD UNIT=SYSDA,DISP=(NEW,PASS),
//          DCB=(RECFM=FB,LRECL=120,BLKSIZE=6000),
//          DSN=&&SCA2,SPACE=(CYL,(4,1))
//SYSPRINT DD SYSOUT=A
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(100),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(100),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(100),,CONTIG)
/*
// PEND
// EXEC SORTC,
&END
&PUNCH // N=&N&
&END
CP SPOOL PUN NOCONT CLOSE
```

PROCEDURA EXEC ADD02

Richiama la procedura MESE, calcola il numero del nastro di input, esegue i filedef necessari manda in esecuzione i programmi ADD02 e SUCC

```
&CONTROL OFF NOMSG
-TY &TYPE MESE E ANNO?
&PEAD ARGS
&IF &INDEX NE 2 &GOTO -TY
EXEC MESE &1 &2
&M = &GLOBAL2
EXEC NASTROAD &M &2
&READ VARS &NA
&TYPE IL NASTRO SULLA UNITA' 181 DEVE ESSERE &NA
-RINDS &TYPE INDIRIZZO DEL DISCO DOVE METTERE LE STAMPE?
&DSK = AAA
&READ VARS &DSK
&IF &DSK = AAA &GOTO -RINDS
ACC &DSK B
&IF &RETCODE NE 0 &GOTO -ERAC
&M1 = GENNAIO
&M2 = FEBBRAIO
&M3 = MARZO
&M4 = APRILE
&M5 = MAGGIO
&M6 = GIUGNO
&M7 = LUGLIO
&M8 = AGOSTO
&M9 = SETTEMBRE
&M10 = OTTOBRE
&M11 = NOVEMBRE
&M12 = DICEMBRE
&I = 1
&LOOP -FSTAT &1 = &M&I
STATE &M&I FAT&2 A
&IF &RETCODE NE 0 &GOTO -NOFP
-FSTAT &I = &I + 1
STATE &1 FAT&2 A
&IF &RETCODE NE 0 &GOTO -OKNE
&TYPE **** ESISTE GIA' IL FILE ' &1 FAT&2 A ' ****
&TYPE LO CANCELLO PER ANDARE AVANTI (SI O NO)?
&SI = SI
&READ VARS &SI
&IF &SI = NO &EXIT 4
ERASE &1 FAT&2 A
-OKNE &I = &I + 1
&LOOP -FST1 &I > 12
STATE &M&I FAT&2 A
&IF &RETCODE EQ 0 &GOTO -NDES
```

```
-FST1 &I = &I + 1
&I = 1
&LOOP 2 12
FI SOLD&I DISK &M&I FAT&2 A (RECFM F BLOCK 26
&I = &I + 1
ERASE COD COD &1
FI FLO1 TAP1 (RECFM FB LRECL 120 BLOCK 6000
FI TABELLE DISK FILE TABELLE * (RECFM F BLOCK 80
FI TOT DISK TOT &1 B (RECFM FA BLOCK 133
FI ADDSUC DISK COD COD A1 (RECFM F BLOCK 4 LRECL 4
FI PROSP1 DISK PROSP1 &1 B (RECFM FA BLOCK 133
FI PROSP2 DISK PROSP2 &1 B (RECFM FA BLOCK 133
FI PROSP3 DISK PROSP3 &1 B (RECFM FA BLOCK 133
FI PROSP4 DISK PROSP4 &1 B (RECFM FA BLOCK 133
FI PROSP5 DISK PROSP5 &1 B (RECFM FA BLOCK 133
FI SOLDI DISK SOLDI ADD02 A (RECFM F BLOCK 26
ACC 194 C/A
&IF &RETCODE NE 0 &GOTO -ERDIS
FI ANAGRAF DISK UTENTI FILE C (RECFM F BLOCK 80 DSORG DA XTENT 8000
CP REW 181
&IF &RETCODE NE 0 &GOTO -NORDY
-CONT LOAD ADD02 TABCOM ADD02FD ADD02IO (NODUP
&IF &RETCODE NE 0 &GOTO -ERL
START
&IF &RETCODE NE 0 &GOTO -EREX
-SSC STATE COD COD A1
&IF &RETCODE NE 0 &GOTO -NCO
FI CODICE DISK COD COD A1 (RECFM F BLOCK 4 LRECL 4
FI ADDEB TAP1 (RECFM FB LRECL 120 BLOCK 6000
FI USC DISK COD1 COD1 A1 (RECFM F BLOCK 120 LRECL 120
CP REW 181
LOAD SUCC (START
ERASE CODICI M10000 A1
REN COD1 COD1 A1 CODICI M10000 A1
-NCO REN SOLDI ADD02 A &1 FAT&2 A
&TYPE I CODICI FHK SONO SUL FILE PROSP1 &1 B
&TYPE I CODICI EIST SONO SUL FILE PROSP2 &1 B1
&TYPE IL RESTO E' SUL FILE PROSP3 &1 B1
&TYPE LE FATTURE NON PAGANTI SONO SUL FILE PROSP4 &1 B1
&TYPE LE FATTURE PAGANTI SONO SUL FILE PROSP5 &1 B1
&EXIT
-ERDIS &TYPE ERRORE &RETCODE IN ACCESSO DISCO 194 ANAG
&EXIT &RETCODE
-EREX &TYPE ERRORE ESECUZIONE PROGRAMMA ADD02
&EXIT &RETCODE
-ERL &TYPE ERRORE LOAD PROGRAMMA ADD02
&EXIT &RETCODE
-ERAC &TYPE ERRORE ACCESSO DISCO &DSK ERR &RETCODE
&EXIT &RETCODE
-NORDY &TYPE NASTRO NON READY QUANDO PRONTO DAI RC.
```



```
&READ  
CP REW 181  
&IF &RETCODE NE 0 &GOTO -NORDY  
&GOTO -CONT  
-NOFP &TYPE ** NON ESISTE FILE MESE PRECEDENTE &M&I FAT&2  
&EXIT  
-NDES &TYPE ** ESISTE FILE DEL MESE SUCCESSIVO &M&I FAT&2  
&EXIT
```