

Indice

Considerazioni sulle scelte del software applicativo usato per la creazione e il mantenimento di una entità contenenti dati relativi all'attività del centro informatico del CNUCE	1
MACRO EXEC SARCHIV	3
SARCHIV AUTO	3
ARCHIV.PROGRAM	3
MVS SAS	3
VM SAS	4
MARC SAS	4
DTSET-OS	5
Descrizione del formato dei dataset MVS	6
Descrizione del formato delle schede VM	7
<u>I listati delle procedure</u>	10
PROCEDURA DI ACQUISIZIONE E ORGANIZZAZIONE DEI DATI DI ATTIVITA' DEL SISTEMA OPERATIVO MVS	10
PROCEDURA DI ACQUISIZIONE E ORGANIZZAZIONE DEI DATI DI ATTIVITA' DEL SISTEMA OPERATIVO VM	15
PROCEDURA DI ACQUISIZIONE E DEI DATI DI UTILIZZO DEI DATASET MVS	20
PROCEDURA DI ACQUISIZIONE E ORGANIZZAZIONE DI DATI DI UTILIZZO DEL PRODOTTO MARC	21
PROCEDURA SARCHIV EXEC	22

**Considerazioni sulle scelte del software applicativo usato per la creazione e il
mantenimento di una entità contenenti dati relativi all'attività del centro
informatico del CNUCE**

Il sistema SAS, un pacchetto software già conosciuto per le sue potenzialità di elaborazioni statistiche e per le sue applicazioni grafiche di buon livello, è stato preso in considerazione anche come strumento adatto alla realizzazione di questo lavoro in quanto si è rivelato un linguaggio di quarta generazione adatto alle nostre esigenze.

Sull'esperienza fatta lavorando con questo prodotto si può di certo affermare che le sue potenzialità non si fermano alla rappresentazione grafica di insiemi di dati e alla statistica ma la facilità di utilizzo si riscontra anche nella scrittura dei programmi applicativi in generale; infatti è molto più limitato il numero di istruzioni necessario alla soluzione di un algoritmo.

Questo è certamente importante per quel che riguarda i successivi interventi di manutenzione e di aggiornamento della procedura applicativa.

Inoltre ha molta importanza il fatto che le procedure stesse prendono una organizzazione strutturata senza che vi sia un grande impegno da parte del programmatore; tutto questo è conseguente al fatto che la sintassi stessa guida quasi automaticamente alla strutturazione delle istruzioni.

Altro punto che ha spinto alla scelta del prodotto in questione è l'agevole e potente gestione degli archivi che nel nostro caso diventa molto importante.

Infatti gli archivi che si creano in relazione ad un determinato insieme di obiettivi vengono fatti confluire in una libreria SAS che li rende di facile identificazione dando praticamente la possibilità di impostare una banca di dati dedicata a successive elaborazioni idonee ad un uso dinamico da parte del software procedurale SAS.

Un'altra delle possibilità che offre questo prodotto che, per noi è di estrema importanza, è la possibilità di intervenire agevolmente su una banca di dati relazionale SQL.

In effetti nel momento in cui il sistema informativo dell'Istituto CNUCE andrà a risiedere su di una banca di dati relazionale i dati dei due archivi "separati" verranno "visti" e trattati dalle procedure come un'entità unica.

Si parla quindi di procedure di accounting che, alimentate dagli archivi sopradetti, andranno a sostituire quelle preesistenti agevolando l'opera di manutenzione ogni volta che si presenteranno problemi nuovi o cambiamenti a livello dei sistemi operativi.

Dall'accounting si passa necessariamente ad un campo applicativo ad esso strettamente legato che diventa essenziale per la conoscenza dell'andamento, la gestione, lo studio di un sistema nella sua attività e nella sua evoluzione.

Si parla, ora, delle valutazioni delle prestazioni, della possibilità di avere a disposizione gli indici principali utili ad un resoconto periodico. E' certamente importante anche il fatto che i dati siano di facile accesso sia grazie alla loro organizzazione, sia per il fatto che procedure a loro dedicate li mettono facilmente a disposizione.

Sempre relativamente a questo argomento va considerato nuovamente il fatto che, data la relativa facilità a impostare procedure grafiche o statistiche, la rilevazione di risultati può diventare estremamente dinamica e non necessariamente legata alle procedure già esistenti.

Di estrema importanza appare il fatto che dal prossimo release "6.7" (distribuito a breve termine) gli archivi in formato SAS potranno essere condivisi tra ambienti diversi (VM, UNIX, DOS) con la conseguente possibilità di distribuzione del lavoro tra sistemi tra loro non compatibili.

Le procedure

- **MACRO EXEC SARCHIV**

Mette a disposizione alle procedure le risorse necessarie all'elaborazione:

 i minidischi su cui risiedono i dati relativi alle schede del VM

 il disco contenente i dati del sistema MVS

 il disco su cui risiedono le informazioni di utilizzo dei datasets MVS

 il disco contenente i dati di utilizzo del prodotto MARC

 avvia il sistema SAS che, in apertura avvia l'autoexec **sarchiv auto**

- **SARCHIV AUTO**

 Richiama dal catalogo **SAS ARCHIV.ADD** il programma seguente

- **ARCHIV.PROGRAM**

 che visualizza un pannello per la richiesta dell'anno e mese che si vuole acquisire.

 In base a questo la procedura chiede all'operatore il montaggio del nastro contenente i dati del periodo richiesto.

 Infine lancia i programmi successivi.

- **MVS SAS**

 E' la prima procedura di acquisizione dati mandata in esecuzione. La fonte dalla quale si estraggono i dati sono i records dell'SMF (System Management Facility) registrati su nastro magnetico (24 nastri: 12 per gli anni pari 12 per quelli dispari). Il nastro viene richiesto, come già detto, dalla procedura ARCHIV PROGRAM.

Nella prima parte della procedura vengono definiti tre data set associati ad una libreria SAS di nome ADDEBITO nella quale confluiranno anche i data set prodotti dalle altre procedure del sistema.

Il type6 e il type26 sono relativi alle informazioni delle stampe reali ed il type30, del quale viene selezionato solo il sottotipo 5 (Job termination) per agli altri dati di accounting. A questo livello vengono selezionati i campi significativi per le finalità che si sono proposte e che saranno sviluppate dalle applicazioni siano esse di rendiconto, addebito, grafiche o statistiche.

La lettura dei records di SMF seleziona i tre records di tipo sopra descritti. I job sono univocamente identificati da tre parametri che costituiscono il "job log identification"; essi si identificano nel *Job name, nel tempo d'inizio e nella data* che, in riferimento ad un job, si trovano in ogni tipo record dell'SMF. Esiste la possibilità, in SAS di leggere i secondi due campi i associazione con un formato particolare "smfstamp8." che rende il contenuto dei due campi come unico valore numerico di successivo più agevole uso.

Va fatto notare che, dato che i formati record sono di tipo variabile bloccato spanned, è necessario, rispetto alla rappresentazione degli offsets che si riscontra sul manuale d'uso di SMF, diminuirli di tre unità.

Al termine della lettura e organizzazione dei nuovi record in formato SAS gli archivi di lavoro creati vengono accodati ai preesistenti archivi che vengono così aggiornati o corretti in qualsiasi giorno dell'anno.

- **VM SAS**

Per quel che riguarda il sistema operativo VM i dati non risiedono su nastri magnetici ma sono registrati su minidisco di macchina virtuale; nel caso specifico i dati dei primi sei mesi sul minidisco con indirizzo virtuale 200 della macchina virtuale AUTOLOG1, i dati dei secondi sei mesi sul minidisco 201.

Le prime istruzioni accedono al minidisco corrispondente alla richiesta nella data del pannello iniziale.

Segue la definizione di quattro data set SAS :

MACVIRT che raccoglie le informazioni dell'attività delle macchine virtuali attive

PAGREAL in cui confluiscono i dati delle stampe reali.

PLOT relativa all'attività del plotter

TDISK registra l'uso, riferito alle macchine virtuali, dei dischi temporanei di lavoro.

- **MARC SAS**

Programma che estrae le informazioni da un disco MVS e registra sulla libreria SAS 'ADDEBITO' le informazioni organizzate come nelle altre procedure.

- **DTSET-OS**

I dati sono estratti dal disco MVS ADDFILE.

Le istruzioni sono rappresentate nel capitolo relativo ai listati dei programmi e l'algoritmo applicato per la realizzazione dei risultati è rappresentato nel seguente schema:

Mese precedente	Mese in esame	Mese successivo
	X.....X-----Y.....Y giorni addebitati	
	X-----Y giorni addebitati	
	X-----Y.....Y giorni addebitati	
X.....X-----Y	X-----Y giorni addebitati	

X= Data di emissione del record

Y= Data della precedente emissione

E' necessario, per avere tutto il mese in esame, che esistano già i dati relativi al mese successivo.

Infatti quei record che hanno il mese di emissione uguale al mese successivo e il mese della precedente emissione uguale al mese in corso verrebbero persi.

Descrizione del formato dei dataset MVS

A) Archivio TYPE6

campo	variabile	descrizione	formato
15-22	JOBNAME	Nome del job	\$8.
23-30	DATASMF	Data-Time	smfstamp8.
85-88	SMF6PGE	Page count	Pib4.

B) Archivio TYPE26

campo	variabile	descrizione	formato
15-22	JOBNAME	Nome del Job	\$8.
23-30	DATASMF	Data-Time	smfstamp8.
57-64	SMF26JID	Job identification	\$8.
65-85	SMF26NAM	Programmer's name	\$20.
105-108	CODICE	Codice dilavoro	\$4.

B) Archivio TYPE30:

Offset	variabile	descrizione	formato
smf30iof	JOBNAME	Nome Job	\$8.
+2	SMF30FNM	JES job identifier	\$8.
+2	SMF30CLS	Classe del job	\$1.
+21	DATASMF	Data-time.....	smfstamp8.
30COF+4	SMF30CPT	Cpu problema in cent. di sec.....	PIB4.
segue	SMF30CPS	Cpu sistema in cent. di sec.....	PIB4.
+8	SMF30JVU	Step vector usage in cent. di sec.....	PIB4.
+4	SMF30JVA	Step vector affinity in cent. di sec	PIB4.

Descrizione del formato delle schede VM

A) Scheda Macchina Virtuale:

ID :

01, C1	Tariffa diurna
31, A1	Tariffa notturna
01, 31	CMS conversazionale
C1, A1	CMS batch

campo	variabile	descrizione	formato
1-8	NOME_MV	Nome macchina virtuale	\$8.
9-16	CODICE	Codice macchina virtuale	\$8.
17-22	DATA	Data di logoff mm/gg/aa	mmddy6.
25-26	MIN	Minuto di logoff	w2.
29-32	TIME_ON	Tempo di ON in secondi	PIB4.
33-36	CPU_TOT	Tempo di CPU totale in millisecondi.	PIB4.
37-40	CPU_VIRT	Tempo di CPU virt in millisecondi ..	PIB4.
73-74	TIP_CPU	Tipo macchina	PK2. *
79-80	ID	Tipo scheda	\$2.

B) Scheda Pagreal:

ID :

05, C5	Tariffa diurna
35, A5	Tariffa notturna
05, 35	CMS conversazionale
C5, A5	CMS batch

campo	variabile	descrizione	formato
1-8	NOME_MV	Nome macchina virtuale	\$8.
9-16	CODICE	Codice macchina virtuale	\$8.

17-22	DATA	Data di logoff mm/gg/aammddyy6.
23-24	ORE	Ora di logoff w2.
25-26	MIN	Minuto di logoff w2.
27-28	SEC	Secondo di logoff w2.
29	FLAG	P=perforazioni S=Stampa\$1.
30-31	I_D	Indirizzo devicePIB2.
32-35	PAGE_PRT	Numero di pagine stampate PIB4.
36-39	RIGHE_PR	Numero totale righe o schede..... PIB4.
42-45	PRT_TIME	Tempo di stampa in secondi PIB4.
79-80	ID	Tipo scheda\$2.

B) Scheda Plot

ID:

77	Tariffa diurna
97	Tariffa notturne

campo	variabile	descrizione	formato
1-8	NOME_MV	Nome macchina virtuale.....	\$8.
9-16	NOME_JOB	Nome job.....	\$8
17-24	CODICE	Codice.....	\$8.
25-28	DATA	Data di esecuzione plot.....	mmddyy4.
29-30	(ORE MIN SEC)	Ora connessione	2.
31-34	DURATA.	Durata esecuzione plot.....	PIB4.
33-36	L_CARTA	Lunghezza carta in cm.	PIB4..

C) Scheda unità dedicate e tdisk

ID:

02, 03, C2, C3	Tariffa diurna
32, 33, A2	Tariffa notturna

campo	Variabile	descrizione	formato
1-8	NOME_MV	Nome macchina virtuale	\$8.
9-16	CODICE	Codice	\$8.
17-22	DATA	Data connessione	mmddyy6.
23-24	(ORE MIN SEC)	Ora di connessione	2.

25-28	CON_TIME	Tempo di connessione PIB4.	
29	CLS_UNIT	Classe di unità	PIB1.
30	TIP_UNIT	Tipo di unità	PIB1.
37-38	NUM_CYL	Numero di cilindri se Tdisk	PIB2.

I listati delle procedure

PROCEDURA DI ACQUISIZIONE E ORGANIZZAZIONE DEI DATI DI ATTIVITA' DEL SISTEMA OPERATIVO MVS

/* GLI ANNI PARI SONO SU: H301 - H312
GLI ANNI DISPARI SONO SU : H313 - H324 */

/****** DETERMINAZIONE DELL'AMBIENTE DI INPUT/OUTPUT *****/

OPTIONS OBS=MAX;
CMS REW 181 ;
FILENAME INPUT TAPE 'TAP1' RECFM=VBS LRECL=30000BLKSIZE=32760;

LIBNAME ADDEBITO 'O';

/****** DEFINIZIONE STEP DI DATA E RELATIVE VARIABILI **/

DATA TYPE6 (KEEP=JOBNAME DATASMF DATA TIME SMF6PGE SISTEMA ANNO

/* SMF6ROUT SMF6SBS SMF6LN2 SMF6CPS SMF6FLC SMF6LN3
SMF6RTE */)

TYPE26 (KEEP=JOBNAME CODICE DATASMF DATA TIME ANNO
SISTEMA SMF26NAM SMF26JID)

TYPE30 (KEEP=TIPO_CPU SMF30TYP
JOBNAME SMF30CLS DATASMF SMF30JNM
SMF30CPT SMF30CPS SMF30JVU SMF30JVA CODICE
DATA TIME SISTEMA SMF30TPR ANNO);

FORMAT DATA DDMMYY8.

TIME SMF30CPT SMF30CPS SMF30JVA SMF30JVU TIME12.2;

LABEL SMF6SBS='SUBSYS ID'

SMF6PGE='PAGINE STAMPATE'

SMF26JID='JES JOB ID'

SMF26NAM='PROGR. NAME'

SMF30TYP='ID SOTTOTIPO'

SMF30JNM='JES JOB ID'

SMF30CLS='JOB CLASS'

SMF30CPT='CPU PROBLEM'

SMF30CPS='CPU SYSTEM'

SMF30JVU='CPU VECTOR USAGE'

SMF30JVA='CPU VECTOR AFF.'

SMF30TPR='NUM. MOUNT';

LENGTH SISTEMA \$4;

/****** INIZIO LETTURA DEI RECORD DI SMF *****/

***** E SELEZIONE DEI TIPI RECORD 6,26 E 30 *****/

INFILE INPUT;

INPUT @2 RECTYPE PIB1. @;

IF RECTYPE='00011110'B | RECTYPE='00000110'B

| RECTYPE='00011010'B;

```
/*----- RECORD TIPO 6 -----*/
IF RECTYPE='0000110'B
  THEN DO;
    SISTEMA='MVS';
    INPUT @15 JOBNAME $8. DATASMF SMFSTAMP8.
      @59 SMF6SBS PIB2.
      @85 SMF6PGE PIB4. @;
    DATA=DATEPART(DATASMF);
    TIME=TIMEPART(DATASMF);
    ANNO=YEAR(DATA);
/**   JES2 OUTPUT WRITER **/
IF SMF6SBS=2
  THEN DO;
    INPUT @89 SMF6RTE PIB2.
      SMF6LN2 PIB2. @;
    IF SMF6LN2^=0
      THEN DO;
        INPUT SMF6CPS PIB8.
          +24 SMF6FLC PIB1.
          +1 SMF6LN3 PIB2. @;
        IF SMF6LN3^=0
          THEN INPUT SMF6ROUT PIB4. @;
        END;
      IF SMF6RTE ='0100'X | SMF6ROUT='00010000'X
        THEN OUTPUT TYPE6;
      ELSE DELETE;
    END;
  END;

/*****      PSF      *****/
IF SMF6SBS=7
  THEN DO;
    INPUT SMF6LN2 PIB2.
      SMF6CPS PIB8.
      +24 SMF6FLC PIB1. @;
    OUTPUT TYPE6;
  END;
END;

/***** RECORD TIPO 26 *****/
IF RECTYPE='00011010'B
  THEN DO;
    INPUT @15 JOBNAME $8. DATASMF SMFSTAMP8.
      @57 SMF26JID $8.
      SMF26NAM $20. @105 CODICE $4. @;
    IF SUBSTR(JOBNAME,1,4)='RSCS' & SMF26NAM='RSCSNET'
      THEN CODICE='H999';
    DATA=DATEPART(DATASMF);
    TIME=TIMEPART(DATASMF);
    ANNO=YEAR(DATA);
    SISTEMA='MVS';
    OUTPUT TYPE26;
  END;

/***** TIPO RECORD 30 *****/
IF RECTYPE='00011110'B
  THEN DO;
    INPUT @11 TIPO_CPU $4.
```

```

@21 SMF30SOF PIB4.
@29 SMF30IOF PIB4. SMF30ILN PIB2. SMF30ION PIB2.
@45 SMF30TOF PIB4. SMF30TIN PIB2. SMF30TON PIB2.
@53 SMF30COF PIB4. SMF30CLN PIB2. SMF30CON PIB2.
@61 SMF30AOF PIB4. SMF30ALN PIB2. SMF30AON PIB2.
@69 SMF30ROF PIB4. SMF30RLN PIB2. SMF30RON PIB2.
@85 SMF30OOF PIB4. SMF30OLN PIB2. SMF30OON PIB2. @;
SMF30SOF+-3;
SMF30IOF+-3;
SMF30TOF+-3;
SMF30COF+-3;
SMF30AOF+-3;
SMF30ROF+-3;
SMF30OOF+-3;
INPUT @SMF30SOF SMF30TYP PIB2.
  @SMF30IOF JOBNAME $8. +24 SMF30JNM $8.
  +2 SMF30CLS $1.
  +21 DATASMF SMFSTAMP8.
  @SMF30COF+4 SMF30CPT PIB4. SMF30CPS PIB4.
  +8 SMF30JVU PIB4. +4 SMF30JVA PIB4.
  @SMF30AOF ACL1 PIB1. ACT1 $VARYING8. ACL1
    ACL2 PIB1. ACT2 $VARYING8. ACL2 @;
  SMF30JNM=SUBSTR (SMF30JNM,1,3);
  IF SMF30JNM='JOB'
    THEN CODICE=ACT2;
  IF SMF30JNM='TSU'
    THEN CODICE=ACT1;
  IF SMF30JNM='STC'
    THEN DELETE;
  IF SMF30JNM='JES'
    THEN DELETE;

/****          TAPE MOUNT          *****/

  IF SMF30OOF ^=0
    THEN DO;
      SMFTAPE=SMF30OOF+8;
      INPUT @SMFTAPE SMF30PTM PIB4.
        SMF30TPR PIB4. @ ;
    END;
/* MOUNT TOT. =      */ SMF30TPR=SMF30PTM+SMF30TPR;
/*****

PER LEGGERE ANCHE I SOTTOTIPI '4' ATTIVARE LA PRIMA RIGHA CHE SEGUE.
E DISATTIVARE LA SECONDA

*****/
/* IF SMF30TYP=5 | SMF30TYP=4; */
  IF SMF30TYP=5;
/*****
  IF SMF30TPR < 0 | SMF30TPR > 99
    THEN SMF30TPR=0;
  IF SMF30CPT < 0
    THEN SMF30CPT=0;
  DATA=DATEPART(DATASMF);
  TIME=TIMEPART(DATASMF);
  ANNO=YEAR(DATA);
  SISTEMA='MVS';

```

```

/*****
RIDUZIONE DEI TEMPI DI CPU DA CENT. DI SEC. A SECONDI
*/
SMF30CPT=SMF30CPT/100;
SMF30CPS=SMF30CPS/100;
SMF30JVU=SMF30JVU/100;
SMF30JVA=SMF30JVA/100;
/*****/
IF VERIFY(SUBSTR(CODICE,1,1),
'ABCDEFGHIJKLMNORSTUVWXYZ')^=0
| VERIFY(SUBSTR(CODICE,2,3),'0123456789')^=0
THEN CODICE=' ';
OUTPUT TYPE30;
END;
RUN;
/*****
FINE LETTURA -- E SORT DEI DATA SET
*****/
PROC SORT DATA=TYPE6;
BY DATASMF JOBNAME;
PROC SORT DATA=TYPE26;
BY DATASMF JOBNAME;
PROC SORT DATA=TYPE30;
BY DATASMF JOBNAME;

PROC SORT DATA=ADDEBITO.TYPE6 FORCE;
BY DATASMF JOBNAME;
PROC SORT DATA=ADDEBITO.TYPE26 FORCE;
BY DATASMF JOBNAME;
PROC SORT DATA=ADDEBITO.TYPE30 FORCE;
BY DATASMF JOBNAME;
/*****
RIDUZIONE DEI DATASET TYPE26 E TYPE30 AD UN RECORD
PER JOB, IN CASO DI DOPPI RECORD
*****/
DATA TYPE26;
SET TYPE26;
BY DATASMF JOBNAME;
IF LAST. JOBNAME
THEN OUTPUT;
RUN;
DATA TYPE30;
SET TYPE30;
BY DATASMF JOBNAME;
IF LAST. JOBNAME
THEN OUTPUT;
RUN;
/*****/
DATA ADDEBITO.TYPE6;
* SET TYPE6;
MERGE ADDEBITO.TYPE6 TYPE6;
BY DATASMF JOBNAME;
IF YEAR(DATA)>YEAR(DATE())-3;
RUN;
DATA ADDEBITO.TYPE26;
* SET TYPE26;
MERGE ADDEBITO.TYPE26 TYPE26;
BY DATASMF JOBNAME;

```

```
IF YEAR(DATA)>YEAR(DATE())-3;
RUN;
DATA ADDEBITO.TYPE30;
* SET TYPE30;
MERGE ADDEBITO.TYPE30 TYPE30;
  BY DATASMF JOBNAME;
IF YEAR(DATA)>YEAR(DATE())-3;
RUN;
```

PROCEDURA DI ACQUISIZIONE E ORGANIZZAZIONE DEI DATI DI ATTIVITA' DEL SISTEMA OPERATIVO VM

```

LIBNAME ADDEBITO 'O';

/*****CONTROLLO SEMESTRI *****/
DATA _NULL_;
MES=SYMGET('M'); M=INPUT(MES,2.);
IF M >=1 & M ^> 6
  THEN CALL SYMPUT ('INPUT',INPUT1);
  ELSE CALL SYMPUT ('INPUT',INPUT2');
RUN

/*****/
DATA MACVIRT (KEEP=TARIFFA TIPOCMS ID NOME_MV CODICE DATA TIME
  CPU_TOT CPU_VIRT PAGE_IN PAGE_OUT TIPO_CPU
  TIME SISTEMA ANNO)
  PAGREAL (KEEP=TARIFFA TIPOCMS ID NOME_MV CODICE DATA FLAG
  PAGE_PRT RIGHE_PR PRT_TIME IND_DEV TIME SISTEMA)
  PLOT (KEEP=TARIFFA TIP_PLOT ID NOME_MV NOME_JOB CODICE
  DATA DURATA L_CARTA TIME SISTEMA ANNO)
  TDISK (KEEP=TARIFFA UNITA ID NOME_MV CODICE DATA CON_TIME
  CLS_UNIT TIP_UNIT TIPOCMS NUM_CYL TIME SISTEMA ANNO);

LENGTH TARIFFA TIPOCMS TIP_PLOT $8 UNITA $6 SISTEMA $4;
ZZ=SYMGET('M');
M=INPUT(ZZ,2.);
INFILE &INPUT;
  INPUT @79 ID $2. @;
  IF ID='01' | ID='31' | ID='C1' | ID='A1'
    | ID='S5' | ID='35' | ID='C5' | ID='A5'
    | ID='77' | ID='97'
    | ID='02' | ID='03' | ID='32' | ID='33'
    | ID='C2' | ID='C3' | ID='A2' | ID='A3';

/***** SCHEDA MACVIRT *****/

IF ID='01' | ID='31' | ID='C1' | ID='A1'
  THEN DO;
  INPUT @1 NOME_MV $8. CODICE $8. DATA MMDDYY6.
  (ORE MIN SEC)(2.) TIME_ON PIB4. CPU_TOT PIB4. CPU_VIRT PIB4.
  PAGE_IN PIB4. PAGE_OUT PIB4. @73 TIP_CPU PK2.;
  ANNO=YEAR(DATA);
  IF M=7 | M=1
    THEN DO;
    IF M=7
      THEN DO;
      IF MONTH(DATA)^=7 & MONTH(DATA) ^=6
        THEN DELETE;
      END;
    IF M=1
      THEN DO;
      IF MONTH(DATA)^=1 & MONTH(DATA) ^=12
        THEN DELETE;

```

```

        END;
    END;
ELSE DO;
    IF MONTH(DATA)^=M
        THEN DELETE;
    END;
TIME=HMS(ORE,MIN,SEC);
TIPO_CPU=PUT(TIP_CPU,4.);
CPU_TOT=CPU_TOT/1000;
CPU_VIRT=CPU_VIRT/1000;
SISTEMA='VM';          /* SISTEMA OPERATIVO */
IF ID='01' | ID='C1'
    THEN DO;
        TARIFFA='DIURNA';
        IF ID='01'
            THEN TIPOCMS='CMS CONV.';
            ELSE TIPOCMS='CMS BATCH';
        END;
    ELSE DO;
        TARIFFA='NOTTURNA';
        IF ID='31'
            THEN TIPOCMS='CMS CONV.';
            ELSE TIPOCMS='CMS BATCH';
        END;
FORMAT DATA DDMMYY. TIME TIME_ON TIME12.
CPU_TOT CPU_VIRT TIME12.2;
OUTPUT MACVIRT;
END;
/***** FINE SCHEDA MACVIRT *****/

/***** SCHEDA PAGREAL *****/
***** STAMPE E PERFORAZIONI *****/
IF ID='S5' | ID='35' | ID='C5' | ID='A5'
    THEN DO;
        INPUT @1 NOME_MV $8. CODICE $8. DATA MMDDYY6.
            (ORE MIN SEC)(2.) FLAG $1. I_D PIB2. PAGE_PRT PIB4.
            RIGHE_PR PIB4. +2 PRT_TIME PIB4.;
        ANNO=YEAR(DATA);
        IF M=7 | M=1
            THEN DO;
                IF M=7
                    THEN DO;
                        IF MONTH(DATA)^=7 & MONTH(DATA) ^=6
                            THEN DELETE;
                    END;
                IF M=1
                    THEN DO;
                        IF MONTH(DATA)^=1 & MONTH(DATA) ^=12
                            THEN DELETE;
                    END;
            END;
        ELSE DO;
            IF MONTH(DATA)^=M
                THEN DELETE;
            END;
        IND_DEV=PUT(I_D,HEX4.);
        TIME=HMS(ORE,MIN,SEC);
        PRT_TIME=PRT_TIME/100; /* TEMPO DI STAMPA IN SECONDI */

```

```
SISTEMA='VM';
IF ID='S5' | ID='C5'
  THEN DO;
    TARIFFA='DIURNA';
    IF ID='S5'
      THEN TIPOCMS='CMS CONV.';
      ELSE TIPOCMS='CMS BATCH';
    END;
  ELSE DO;
    TARIFFA='NOTTURNA';
    IF ID='35'
      THEN TIPOCMS='CMS CONV.';
      ELSE TIPOCMS='CMS BATCH';
    END;
  FORMAT DATA DDMMYY. TIME TIME8.;
  OUTPUT PAGREAL;
END;
/***** FINE SCHEDA PAGREAL *****/

/***** SCHEDA PLOT *****/
IF ID='77' | ID='97'
  THEN DO;
    INPUT @1 NOME_MV $8. NOME_JOB $8. CODICE $8. DATA MMDDYY6.
      (ORE MIN SEC) (2.)
      DURATA PIB4. /* TEMPO ESECUZIONE PLOT IN SECONDI*/
      L_CARTA PIB4. /* LUNGHEZZA CARTA IN CM. */
      @55 TIP $1.;
    ANNO=YEAR(DATA);
    IF M=7 | M=1
      THEN DO;
        IF M=7
          THEN DO;
            IF MONTH(DATA)^=7 & MONTH(DATA) ^=6
              THEN DELETE;
            END;
          IF M=1
            THEN DO;
              IF MONTH(DATA)^=1 & MONTH(DATA) ^=12
                THEN DELETE;
              END;
            END;
          ELSE DO;
            IF MONTH(DATA)^=M
              THEN DELETE;
            END;
          TIME=HMS(ORE,MIN,SEC); /* ORA ESEC. PLOT */
          SISTEMA='PLOT';
          IF TIP='C'
            THEN TIP_PLOT='CALCOMP';
            ELSE TIP_PLOT='VERSATEC';
          IF ID='77'
            THEN TARIFFA='DIURNA';
            ELSE TARIFFA='NOTTURNA';
          FORMAT DATA DDMMYY. TIME TIME12.;
          OUTPUT PLOT;
        END;
      /***** FINE SCHEDA PLOT *****/
```

```

/***** SCHEDA UNITA' DEDICATE E TDISK *****/
IF ID='02' | ID='03' | ID='32' | ID='33' | ID='C2' | ID='C3' | ID='A2'
 | ID='A3'
THEN DO;
  INPUT @1 NOME_MV $8. CODICE $8. DATA MMDDYY6.
    (ORE MIN SEC) (2.)
  CON_TIME PIB4. /* TEMPO DI CONNESSIONE IN SEC. */
  CLS_UNIT PIB1. /* CLASSE DELL' UNITA' */
  TIP_UNIT PIB1. /* TIPO UNITA' */
  @37 NUM_CYL PIB2.; /* NUMERO CYL. SE TDISK*/
ANNO=YEAR(DATA);
IF M=7 | M=1
THEN DO;
  IF M=7
  THEN DO;
    IF MONTH(DATA)^=7 & MONTH(DATA) ^=6
    THEN DELETE;
  END;
  IF M=1
  THEN DO;
    IF MONTH(DATA)^=1 & MONTH(DATA) ^=12
    THEN DELETE;
  END;
END;
ELSE DO;
  IF MONTH(DATA)^=M
  THEN DELETE;
END;
IF (ID='02' | ID='32' | ID='C2' | ID='A2') & CLS_UNIT^=8
THEN DELETE;
IF (ID='03' | ID='33' | ID='C3' | ID='A3') & CLS_UNIT^=4
THEN DELETE;
IF ID='02' | ID='03' | ID='C2' | ID='C3'
THEN DO;
  TARIFFA='DIURNA';
  IF ID='02' | ID='03'
  THEN TIPOCMS='CMS CONV.';
  ELSE TIPOCMS='CMS BATCH';
END;
ELSE DO;
  TARIFFA='NOTTURNA';
  IF ID='32' | ID='33'
  THEN TIPOCMS='CMS CONV.';
  ELSE TIPOCMS='CMS BATCH';
END;
TIME=HMS(ORE,MIN,SEC); /* ORA DI ADDEBITO */
SISTEMA='VM';
IF SUBSTR(ID,2,1)='2'
THEN UNITA='NASTRO';
ELSE UNITA='TDISK';
FORMAT DATA DDMMYY. TIME TIME12.;
OUTPUT TDISK;
END;
RUN;

/** FINE SCHEDA TDISK **/

/***** ORDINAMENTO PER IL MERGE SUCCESSIVO *****/

```

```
PROC SORT DATA=MACVIRT;
  BY DATA TIME NOME_MV ID;
PROC SORT DATA=PAGREAL;
  BY DATA TIME NOME_MV ID;
PROC SORT DATA=PLOT;
  BY DATA TIME NOME_MV ID;
PROC SORT DATA=TDISK;
  BY DATA TIME NOME_MV ID;
```

```
PROC SORT DATA=ADDEBITO.MACVIRT FORCE;
  BY DATA TIME NOME_MV ID;
PROC SORT DATA=ADDEBITO.PAGREAL FORCE;
  BY DATA TIME NOME_MV ID;
PROC SORT DATA=ADDEBITO.PLOT FORCE;
  BY DATA TIME NOME_MV ID;
PROC SORT DATA=ADDEBITO.TDISK FORCE;
  BY DATA TIME NOME_MV ID;
```

```
/******
      AGGIORNAMENTO ARCHIVIO
(PER CREAZIONE USARE IL SET, POI COMMENTARLO E LIBERARE IL MERGE)
*****/
```

```
DATA ADDEBITO.MACVIRT;
* SET MACVIRT;
MERGE ADDEBITO.MACVIRT MACVIRT;
BY DATA TIME NOME_MV ID;
IF YEAR(DATA)>YEAR(DATE())-3;
IF LAST.ID
  THEN OUTPUT;
RUN;
DATA ADDEBITO.PAGREAL;
* SET PAGREAL;
MERGE ADDEBITO.PAGREAL PAGREAL;
BY DATA TIME NOME_MV ID;
IF YEAR(DATA)>YEAR(DATE())-3;
IF LAST.ID
  THEN OUTPUT;
RUN;
DATA ADDEBITO.PLOT;
* SET PLOT;
MERGE ADDEBITO.PLOT PLOT;
BY DATA TIME NOME_MV ID;
IF YEAR(DATA)>YEAR(DATE())-3;
IF LAST.ID
  THEN OUTPUT;
RUN;
DATA ADDEBITO.TDISK;
* SET TDISK;
MERGE ADDEBITO.TDISK TDISK;
BY DATA TIME NOME_MV ID;
IF YEAR(DATA)>YEAR(DATE())-3;
IF LAST.ID
  THEN OUTPUT;
RUN;
```

PROCEDURA DI ACQUISIZIONE E DEI DATI DI UTILIZZO DEI DATASET MVS

```

/***** CREAZIONE DATASET OS *****/

LIBNAME ADDEBITO 'O';
OPTIONS OBS=MAX;
DATA C;
INFILE DTSETIN;
INPUT @1 CODICE $4. ANNO_E 2. GIORNI_E 3. ANNO_P 2. GIORNI_P 3.

      NBLOCK 3. PROD 6. NOMEVOL $6. NOMEDS $44.;
      MESE=INPUT(SYMGET('M'),2.);
      ANNO=INPUT(SYMGET('A'),4.);
M=MESE+1;
M2=MESE-1; /* MESE PRECEDENTE AL MESE IN CAUSA */
A2=ANNO;
A=ANNO;
IF MESE=1
  THEN DO;
    M2=12;
    A2=ANNO-1
  END;
IF MESE=12
  THEN DO;
    M=1;
    A=ANNO+1;
  END;
  START1= MDY(M2,1,A2); /* PRIMO GIORNO MESE
PRECEDENTE */
  START = MDY(MESE,1,ANNO); /* PRIMO GIORNO MESE IN CAUSA */
  STOP = MDY(M,1,A)-1; /* ULTIMO GIORNO MESE IN CAUSA */
  DATA=MDY(1,1,ANNO_E)+(GIORNI_E-1);
  DATAPREC=MDY(1,1,ANNO_P)+(GIORNI_P-1);
  IF DATA >= START;
  IF DATAPREC <= STOP;
  IF DATA > STOP & DATAPREC > STOP
    THEN DELETE;
  IF DATAPREC < START1
    THEN DELETE
FORMAT START1 START STOP DATA DATAPREC DDMMYY8.;

/*****
* PROC PRINT;
* VAR DATA CODICE DATAPREC NOMEVOL NOMEDS;
PROC SORT DATA=C;
  BY NOMEVOL CODICE NOMEDS;
DATA C;
KEEP
  CODICE PRODOTTO NOMEVOL NOMEDS DATA GIORNI NBLOCK;
  FORMAT DATA DATAPREC DDMMYY8. ;
  SET C;
  BY NOMEVOL CODICE NOMEDS;
  IF FIRST.NOMEDS

```

```
        THEN DO;
            PRODOTTO=0;
            GIORNI=0;
        END;
    IF DATAPREC <= START & DATA >= STOP
    THEN DO;
        G=STOP-START;
        PRODOTTO+(G*NBLOCK);
        GIORNI+G;
    END;
    IF DATAPREC > START & DATA < STOP
    THEN DO;
        G=DATA-DATAPREC;
        PRODOTTO+(G*NBLOCK);
        GIORNI+G;
    END;
    IF DATAPREC > START & DATA >= STOP
    THEN DO;
        G=STOP-DATAPREC;
        PRODOTTO+(G*NBLOCK);
        GIORNI+G;
    END;
    IF DATAPREC <= START & DATA < STOP
    THEN DO;
        G=DAY(DATA);
        PRODOTTO+(G*NBLOCK);
        GIORNI+G;
    END;
    IF DATA=START
    THEN DO;
        PRODOTTO+-NBLOCK;
        GIORNI+-1;
    END;
    IF LAST.NOMEDS
    THEN DO;
        DATA=STOP; /* FORZATA ALL'ULTIMO GIORNO DEL MESE */
        ANNO=YEAR(DATA);
        OUTPUT; /* PER PERMETTERE LA SOVRAPPOSIZIONE */
        END; /* DI SUCCESSIVE LETTURE DELLO STESSO */
            /* MESE SULLO STESSO RECORD */
    PROC SORT DATA=C;
        BY DATA CODICE NOMEVOL NOMEDS;
    PROC SORT DATA=ADDEBITO.DATASET FORCE;
        BY DATA CODICE NOMEVOL NOMEDS;
    DATA ADDEBITO.DATASET;
    * SET C;
        MERGE ADDEBITO.DATASET C;
        BY DATA CODICE NOMEVOL NOMEDS;
    IF YEAR(DATA)>YEAR(DATE()-3); RUN;
```

PROCEDURA DI ACQUISIZIONE E ORGANIZZAZIONE DI DATI DI UTILIZZO DEL PRODOTTO MARC

```
LIBNAME ADDEBITO 'O';
```

```

OPTIONS OBS=MAX;
DATA ADDEBITO.MARC;
  LENGTH SISTEMA $4;
  SISTEMA='MARC';
  TIME=.;
  FORMAT SMF30CPT TIME12.2 DATA DDMMYY8.;
  INFILE MARC;
  INPUT @2 JOBNAME $8. @16 CODICE $4. @21 SMF30CPT 6.
    @28 DATA MMDDYY8.;
  IF YEAR(DATA)>YEAR(DATE())-3;
  ANNO=YEAR(DATA);
RUN;

```

PROCEDURA SARCHIV EXEC

```

/* */
TRACE O
ADDRESS COMMAND
ERASE ARCHIV LOG A
'FI * CLEAR'
'ACC 194 O'
'REL B (DET'
'REL C (DET'
'EXEC LINK AUTOLOG1 200 200 RR ACCT200R'
'EXEC LINK AUTOLOG1 201 201 RR ACCT201R'
'ACC 200 B '
'ACC 201 C '
'EXEC VSAC LIBSVS E '
'FI DTSETIN E DSN MVS ADDFILE '
'FI MARC E DSN MARC ACCT '
'FI INPUT TAP1 (RECFM VBS LRECL 30000 BLKSIZE 32760 '
'FI INPUT1 DISK ACNT DATA B '
'FI INPUT2 DISK ACNT DATA C '

"EXEC SAS (AUTOEXEC='SARCHIV AUTO A' WORK='O' "
"REL B(DET"
"REL C(DET"
"REL E(DET"
/* 'FI * CLEAR' */

```

PROCEDURA DI IMMISSIONE SCELTA , SELEZIONE NASTRI E LANCIO PROCEDURE SUCCESSIVE

DIGITARE ANNO E MESE

anno : &A__

mese : &M

```
&MSG _____
/*SOURCE*/

init:
rc=libname('addebito', 'o');
return;
main:
cursor a;
msg=' ';
refresh;
call symput('a',a);
call symput('m',m);
g=day(date());
if (m=4 | m=6 | m=9 | m=11) & g>30
  then g=30;
if m=2 & g>28
  then g=28;
dat= mdy(m,g,a); /* data richiesta */
if dat>date()
  then do;
    msg='Attenzione!! data posteriore a quella attuale';
    return;
  end;
a1=year(date())-1;
datold= mdy(7,1,a1); /* 01/07/anno precedente */
if m <=6 & dat < datold
  then do;
    msg='Attenzione!! data inferiore a luglio '|| a1;
    return;
  end;
x=mod(a,2);
if x=0
  then
  select (m);
    when(1) call symput('n','H301');
    when(2) call symput('n','H302');
    when(3) call symput('n','H303');
    when(4) call symput('n','H304');
    when(5) call symput('n','H305');
    when(6) call symput('n','H306');
    when(7) call symput('n','H307');
    when(8) call symput('n','H308');
    when(9) call symput('n','H309');
    when(10) call symput('n','H310');
    when(11) call symput('n','H311');
    when(12) call symput('n','H312');
  end;
if x^=0
  then select (m);
    when(1) call symput('n','H313');
    when(2) call symput('n','H314');
    when(3) call symput('n','H315');
    when(4) call symput('n','H316');
    when(5) call symput('n','H317');
    when(6) call symput('n','H318');
    when(7) call symput('n','H319');
```

```
        when(8) call symput('n','H320');
        when(9) call symput('n','H321');
        when(10) call symput('n','H322');
        when(11) call symput('n','H323');
        when(12) call symput('n','H324');
    end;
    alarm;
    msg='Esecuzione in corso';
    refresh;
    submit continue;
    * data;
    * a=symget('a');
    * m=symget('m');
    * c=symget('n');

/* MVS */

    cms nastro mount &n(pass smf;
    cms tape rew;

    filename mvs 'mvs sas a';
    %include mvs;

/* VM */

    filename vms 'vm sas a';
    %include vms;

/* Data set mvs */

    filename dtset 'DTSET_OS SAS A';
    %include dtset;

/* MARC */

    filename mar 'MARC SAS A';
    %include mar;

run;
endsubmit;
alarm;
msg='Esecuzione avvenuta';

return;

term:
    submit;
    cms exec nastro demount;
        dm 'bye';
    endsubmit;
return;
```

PROCEDURA AUTOEXEC SAS PER IL LANCIO

DEI PROGRAMMI SAS

```
DM 'AF C=ARCHIV.ADD.ARCHIV.PROGRAM' AF;  
DM LOG 'FILE "ARCHIV LOG A" LOG';  
DM 'RESET';
```

