

Consiglio Nazionale delle Ricerche

**Informazione sullo stato dei jobs
nei sistemi interconnessi**

370/158 - 370/168

G. Bartoli-F. Leonardi - P. Mogorovich

170

CNUCE

A cura di: G. Bartoli
F. Leonardi
P. Mogorovich

Copyright - Giugno 1980

by - CNUCE - Pisa

Istituto del Consiglio Nazionale delle Ricerche

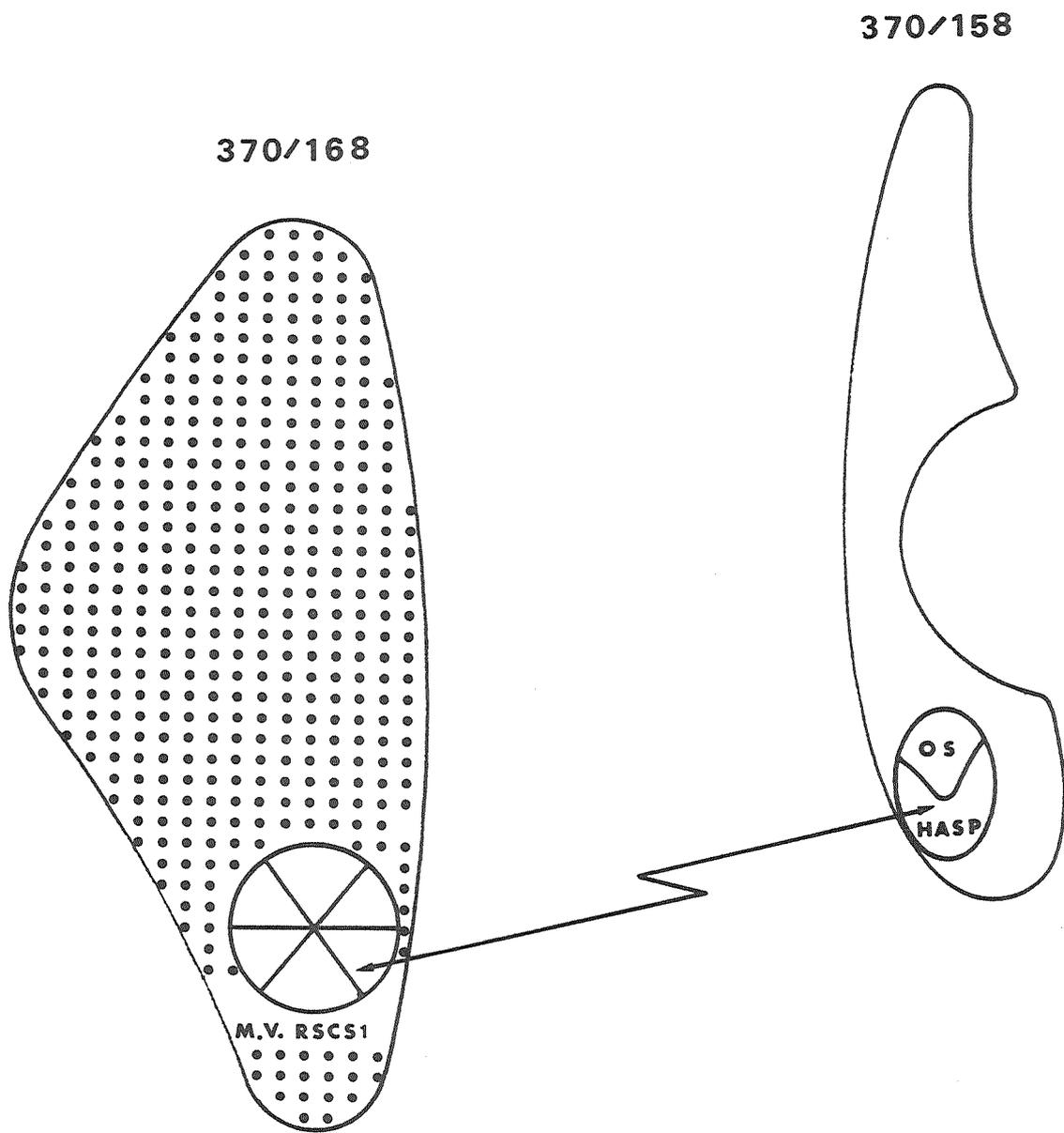
INTRODUZIONE

Il CNUCE, per poter svolgere la sua attivita' di servizio nei confronti di un'utenza di tipo scientifico, e' dotato di due elaboratori: un IBM 370/158 ed un IBM 370/168.

Al fine di offrire un servizio quanto piu' possibile completo ed efficiente, le due macchine sono specializzate ciascuna con un orientamento ben preciso, ed operano gestite da Sistemi Operativi diversi.

Il sistema IBM 370/158 e' gestito da un Sistema Operativo OS VS2/R1.7-HASP. L'altro con un Sistema VM Release 5.

La prima delle due macchine svolge un'attivita' tipicamente 'batch', con 9 partizioni attive, e non ha alcun utente conversazionale (TSO). L'altra macchina svolge, per la massima parte, un servizio interattivo, con un numero di utenti tipico di oltre 100. I due elaboratori sono connessi tramite una linea ad alta velocita', gestita dalla parte del /158 (VS2) come la linea di un remoto di HASP, e dalla parte del /168 (VM) come linea 'attaccata' ad una macchina virtuale, di nome RSCS1, su cui e' attivo, oltre ad altri, un software che simula un remoto di HASP. (fig. 1)



Schema di connessione tra i due elaboratori

FIG. 1

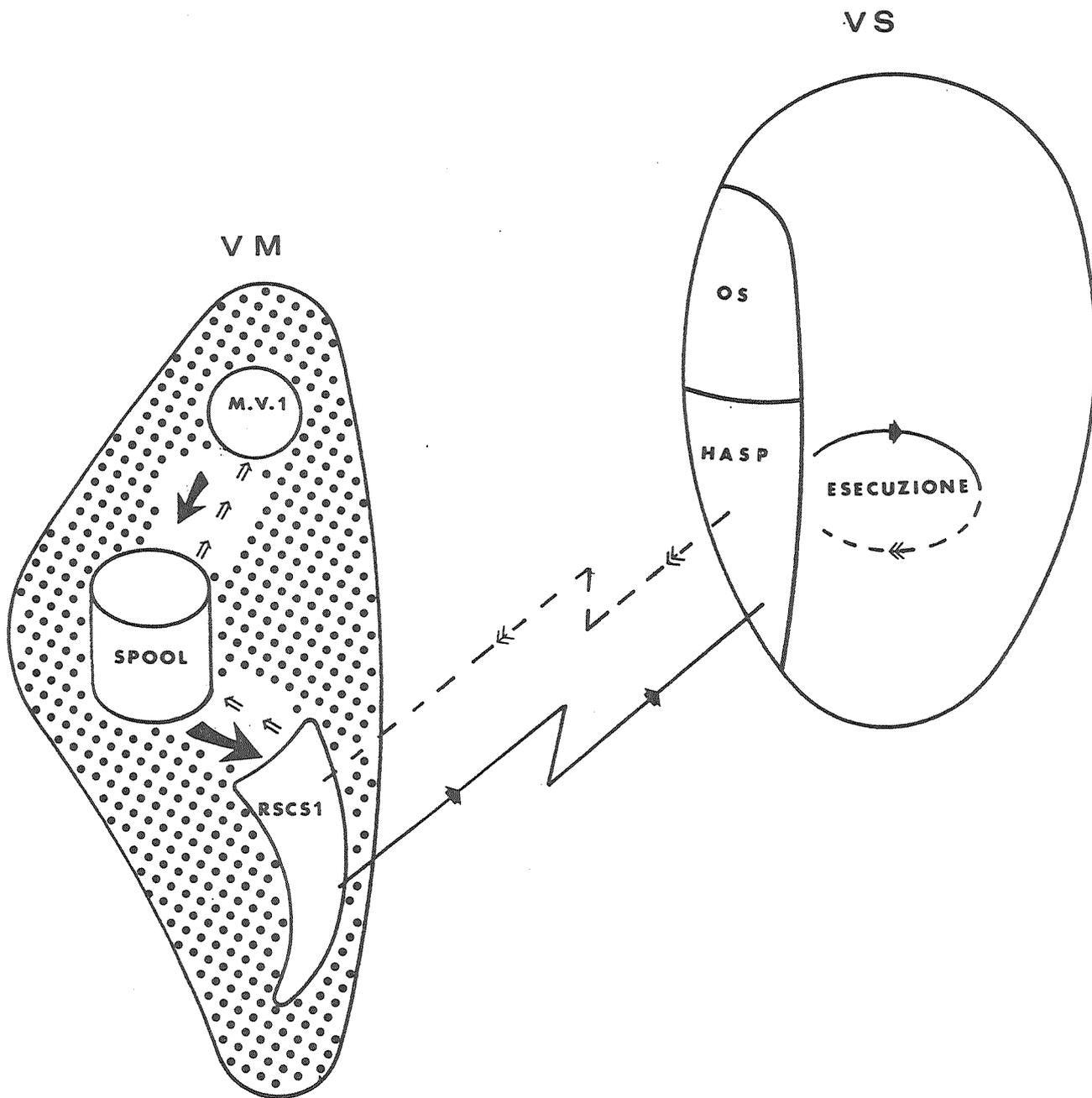
Dalla connessione delle due macchine, e conoscendo le possibilita' offerte dal VM circa lo scambio di 'files' tra macchine virtuali, e' evidente come sia possibile, per qualsiasi utente su una macchina virtuale inviare un file (tipicamente un job) alla macchina virtuale RSCS1, la quale lo inviera' al /158. Una volta eseguito, il job verra' inviato indietro, sulla stessa linea, alla macchina virtuale RSCS1, la quale, a sua volta, lo girera' al mittente originario (fig. 2). Il corretto movimento di files e' ottenuto tramite modifiche ai Sistemi Operativi di RSCS1 e di HASP.

Questo modo di operare da parte degli utenti e' divenuto comunissimo, anche in considerazione della loro dislocazione geografica. Infatti non e' piu' necessario accedere ai locali della sala macchine per il caricamento dei pacchi di schede ed il ritiro delle stampe.

L'utente che opera in questo modo, pero', si trova nella impossibilita', una volta inviato un job, di verificarne lo stato. Cio' comporta uno spreco di risorse, in quanto, talvolta, l'utente dopo un certo tempo di attesa invia di nuovo lo stesso job, supponendo che, per qualche errore, il job precedente non sia stato

eseguito, oppure, altre volte, attende un risultato mentre il job inviato, per una qualsiasi ragione, non e' stato eseguito correttamente.

Da parte di piu' utenti era stata, di conseguenza, richiesta la possibilita' di poter avere informazioni, da un terminale di macchina virtuale sotto VM, di un job presumibilmente in esecuzione sotto VS, sull'altra macchina.



movimento di job

FIG. 2

APPROCCI AL PROBLEMA

Un primo tentativo di risolvere il problema era stato fatto pensando di inviare informazioni lungo la linea di comunicazione (fig. 2), riguardanti lo stato dei job. Questo tipo di soluzione richiedeva modifiche al software di RSCS1 e di HASP, probabilmente non elementari: infatti, nel caso di due o più richieste di informazioni, non era possibile associare le varie risposte a ciascuna richiesta. Si aveva inoltre un appesantimento non trascurabile sulla linea di comunicazione, linea usata in genere piuttosto pesantemente.

Una seconda idea, rivelatasi facilmente attuabile, è la seguente: il parco dischi con cui operano i due elaboratori è collegato ad ambedue i sistemi: ciò vuol dire che un qualsiasi disco in uso da parte del sistema /158, è accessibile anche dall'altro sistema.

Vediamo ora come è organizzato il disco magnetico usato da HASP per lo 'SPOOLING'. Esso è formato, oltre che dalla VTOC e da qualche 'dataset' normalmente non usato, da un'unico DATASET, di nome SYS1.HASPACE, che

occupa praticamente tutto il disco. Questo dataset , a sua volta, e' formato da una serie di records, opportunamente organizzati, e da 3 records speciali. Questi ultimi, situati all'inizio del dataset, vengono usati da HASP per fare un 'CHECKPOINT' periodico e frequente di tabelle che rispecchiano lo stato dei vari jobs. Queste tabelle sono aggiornate ogni volta che un job cambia stato (tab. 3), e, ad ogni aggiornamento, sono scritte sul discc di SPOOL, inoltre una scrittura avviene ogni 60 secondi a causa di aggiornamenti di minore importanza.

Per quanto detto sopra queste tabelle sono accessibili da un utente di macchina virtuale sotto VM, tramite l'invio del comando 'STATO' come piu' avanti specificato.

CAMBIAMENTI DI STATO ORDINARI

STATO INIZIALE

STATO FINALE

NON PRESENTE.....	ON READER
ON READER.....	AWAITING EXEC
AWAITING EXEC.....	EXECUTING
EXECUTING.....	AWAITING OUT
AWAITING OUT.....	AWAITING PURGE
AWAITING PURGE.....	NON PRESENTE

TAB. 3

CONSIDERAZIONI SU UN POSSIBILE DEGRADAMENTO

Un'importante considerazione riguarda l'interferenza della procedura 'STATO' sotto VM con l'algoritmo usato da HASP per gestire il proprio 'SPOOLING'.

Nonostante il disco magnetico sia un supporto tipicamente non sequenziale, possiamo definire 'distanza' tra due records un numero proporzionale al movimento del braccio che guida le testine magnetiche per raggiungere il secondo record, partendo dal primo. Poiche' il tempo di accesso di un record dipende dal movimento suddetto l'organizzazione dei records su un disco puo' influenzare pesantemente le prestazioni di un sistema.

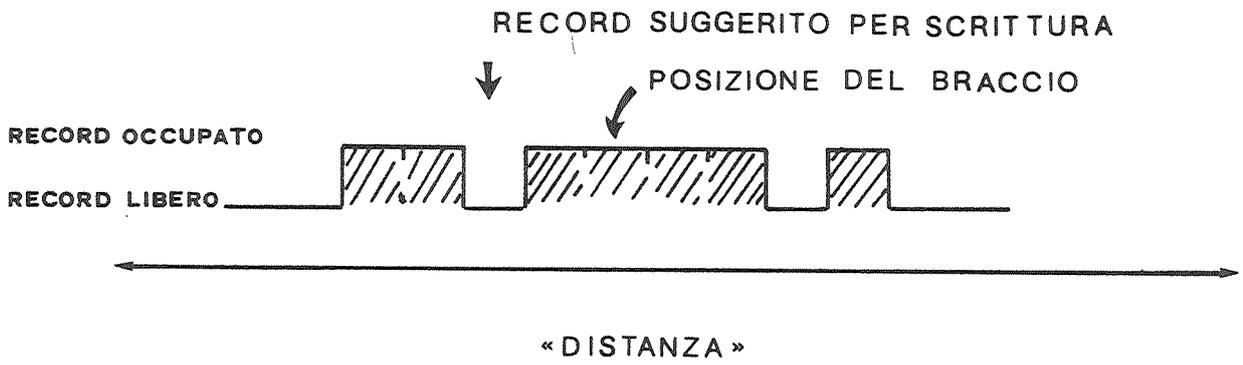
HASP usa uno speciale algoritmo per ottimizzare le sue prestazioni. Tale algoritmo, oltre ad altre funzioni, informa HASP, se si ha una richiesta di scrittura, su quale e' il record libero piu' vicino all'attuale posizione del braccio porta-testine (fig. 4). Per 'attuale posizione del braccio', perc', non si intende la reale posizione fisica, sconosciuta ad HASP, ma la posizione dove HASP ha posizionato il braccio

nell'ultima operazione di I/O. L'algoritmo e' efficace, ovviamente, nel caso di utilizzo esclusivo del disco da parte di HASP. Supponiamo che una operazione di I/O abbia portato il braccio in una posizione (fig. 5) indicata simbolicamente col numero 6, e supponiamo che tale operazione sia stata compiuta da HASP. Supponiamo ancora che una operazione di I/O non dovuta ad HASP abbia portato il braccio in posizione 18. Si ha ora da parte di HASP una richiesta di scrittura e l'algoritmo di ottimizzazione spiegato, indica come record da usare il numero 5: HASP scrive correttamente quanto deve nel record numero 6, ma ccssi' facendo ha mosso il braccio assai piu' di quanto era necessario.

Questo semplice esempio fa capire come sia fondamentale non interferire con l'attivita' di I/O del disco ove HASP compie il suo 'SPOOLING'.

Vediamo adesso se il comando 'STATO' influisce sensibilmente sulle prestazioni dello SPOOLING di HASP.

Osserviamo prima di tutto quanto segue. Valutazioni sul carico di HASP (circa 500 jobs in 10 ore) dicono che si hanno circa 5 scritture di records di checkpoint ogni



come HASP ottimizza lo spool

FIG. 4

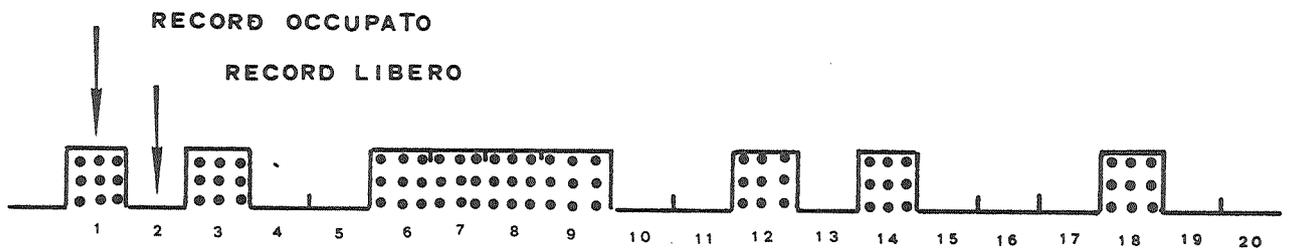
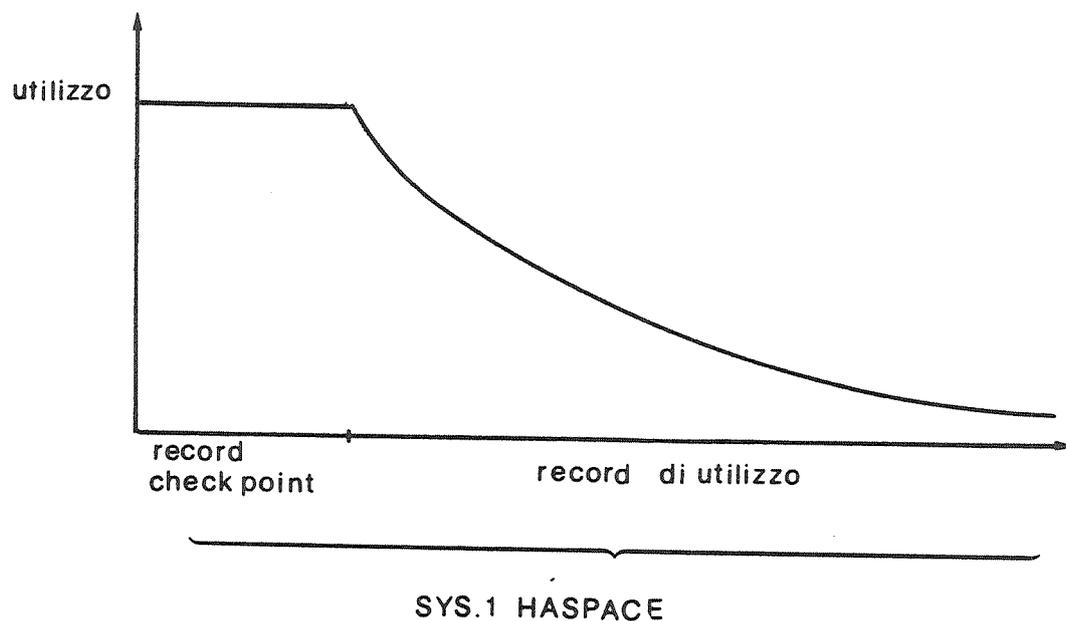


FIG. 5

minuto. Cio' significa che circa ogni 10 secondi il braccio porta testine e' trascinato in una ben determinata posizione, e da li', ogni volta parte il meccanismo di ottimizzazione della scrittura. Quindi i records vicini alla suddetta posizione sono praticamente sempre usati, e l'uso e' via via meno frequente man mano che da tale posizione ci si allontana. Quanto detto e' confermato da una serie di misure effettuate, ed il grafico di utilizzo del DATASET SYS1.HASPACE risulta qualitativamente quello indicato in fig. 6. Lo stesso grafico, se normalizzato, indica anche la percentuale (o se si preferisce la probabilita') di presenza del braccio nelle varie posizioni.

L'utilizzo del comando 'STATO', come detto, prevede la lettura di 3 records di CHECKPOINT, e nel fare questo riporta il braccio porta-testine nella posizione dove e' piu' probabile che esso risieda. In altre parole se dopo le operazioni di I/O dovute al comando 'STATO' si richiede una scrittura, con l'intervento dell'algoritmo piu' volte citato, e' molto probabile che l'algoritmo stesso commetta un errore piccolo, in quanto il braccio si trova staticamente vicino a dove HASP crede che sia.

Queste considerazioni, anche se non suffragate da un'analisi quantitativa, fanno supporre che la degradazione temuta sia quasi trascurabile.



Utilizzo record del disco di spool

FIG. 6

PROCEDURE OPERATIVE

L'utilizzo del comando 'STATC' da parte di un utente di macchina virtuale VM puc' avviene in tre forme:

1) L'invio del comando 'STATO', seguito dopo un blank da una stringa di caratteri, causa la scrittura a terminale delle informazioni relative al job il cui nome e' espresso dalla stringa stessa.

2) Inviando il comando 'STATO' senza alcun parametro, il programma chiede a terminale il nome del job di cui si vogliono le informazioni. Scritte tali informazioni, il programma resta in attesa di un nuovo jobname, e cosi' via, finche' l'invio di un 'ritorno carrello' a vuoto causa la fine del programma.

Da notarsi, che la lettura delle tracce di CKPT avvengono solamente quando si invia il comando STATO: quindi le informazioni su piu' jobs con diverso JOENAME in modo conversazionale sono da ritenersi riferite all'esatto momento in cui parte il comando 'STATO'.

Nel caso vi siano piu' JOBS con il nome uguale a quello richiesto, vengono fornite le informazioni relative a tutti i JOBS con quel nome.

3) Inviando il comando 'STATO' seguito da un punto interrogativo causa la scrittura di informazioni sull'uso del comando stesso.

RISPOSTA DEL COMANDO STATO

Vediamo adesso ccn esempi l'usc e le possibili informazioni che vengono scritte a terminale dopo l'invio del comando:

STATO PROVA1

in risposta

PROVA1 JOB NOT FOUND

In questo caso, il job di nome PROVA non e' sul sistema VS.

STATO PROVA

in risposta

```

  --1--  --2--  --3--  --4--  --5--  --6--
|  |  |  |  |  |  |  |  |  |  |  |
JOB 215 PROVA AWAIT EXEC PRIO 7 CLASS G CODICE K211

  --7--  --8--
|  |  |  |  |
FORM STD. LOCAL

  -----9-----  -----10-----
|  |  |  |  |  |  |  |  |  |  |  |
MSG 'P142 noring, discc PREN01' ==> TPH9 1 SPECIAL <==
```

dove:

- 1) NUMERO DEL JOB
- 2) NOME DEL JOB
- 3) STATO DEL JOB, che puo' essere:

ON HEADER : Il job e' in lettura : in questo caso
non avremo le informazioni dal n. 4 al
n. 10

AWAIT EXEC : Il job e' in attesa di esecuzione

EXECUTING : Il job e' in esecuzione

AWAIT OUT : Il job e' in attesa di stampa, oppure
e' in fase di stampa

- 4) PRIORITA' (di esecuzione o di stampa)
- 5) CLASSE DI ESECUZIONE
- 6) CODICE DI ADDEBITO
- 7) FORMATO con cui vogliamo sia eseguita la stampa
- 8) STAMPA : LOCAL per stampa in Sala Macchine con
le stampanti del /158
: REMOTO per stampa sulla linea di
comunicazione di RSCS1
- 9) MESSAGE : Tra apici compare, per una lunghezza
di 32 caratteri, l'ultima /*MESSAGE
del JOB, unica richiamabile
dall'operatore VS: si puo' quindi
controllare l'esattezza delle
informazioni fornite a quest'ultimo
per la esecuzione del JOB.
- 10) DEVICES : Tra ==> <== compare la richiesta di

unita' fatta tramite le scheda
/*SETUP, ed in particolare:

TPV9 n numero unita' nastri a 9 piste
densita' 800/1600

TPH9 n numero unita' a 9 piste densita'
1600/6250

SPECIAL puo'essere: unita' a 7 piste, nastri
per plotter o dischi

USO DEL COMANDO STATO IN CONVERSAZIONALE

(fare riferimento a quanto specificato nelle PROCEDURE
OPERATIVE)

STATO

ENTER JOENAME ==>

PROVA

PROVA JOB NCT FCUND

ENTER JOENAME ==>

PROVA1

(informazioni sul job PROVA1)

ENTER JOENAME ==>

XYZ

(informazioni sul job XYZ)

ENTER JOENAME ==>

(per terminare, RITORNC A VUOTO)

IL PROGRAMMA NON FORNISCE VOLUTAMENTE INFORMAZIONI SULLO
STATO DI HOLD DEL JOE

STAMPATO PRESSO IL
SERVIZIO TECNOGRAFICO
DEL CNUCE