Network Working Group
Request for Comments: 477
NIC: 14922
References: RFCs 354, 407,
NIC 16306

M. Krilanovich
UCSB

23 May 1973

Remote Job Service at UCSB

Introduction

This RFC is the follow-on document to RFC #436, which briefly described UCSB'S RJS. This document restates the essence of the official RJE protocol (RFC #407), and documents in detail UCSB's implementation of it.

The program described here is available under socket 5 at UCSB, and supports a subset of the official protocol. Specifically, no checks are made for RJE control cards in the input file, jobs may not be batched in the input file, only output file dispositions of discard and transmit-and-discard are implemented, no restart markers are sent on output in FTP blocked mode, and several of the commands are not implemented. There are also other ways in which RJS is known to be less than ideal. For example, whenever any error is detected while processing a job, such as the output's destination host being dead, the job is abandoned, and any further output deleted. A re-write of RJS is scheduled for the near future (in about six months), and many of these deficiencies will be corrected at that time. (Any suggestions for further improvements are more than welcome, and may be made through the Journal to MCK, by mail to the author at the UCSB Computer Center, or by telephone at (8O5) 961-3454.)

In addition to the deviations from the protocol stated above, several modifications have been made to increase user convience. Specifically, the INACCT, OUTACCT, and ACCT commands have been added to accommodate users of TENEX and other systems requiring an account number, OUTPATH has been added as a synonym for OUT and INUSER for INID, and the BYE command does not cause an ABORT. Also, in recognition of the fact that the official protocol is biased heavily toward use by programs, and is therefore rather cumbersome for human users, an alternate, optional, command syntax has been provided. An attempt was made to make this alternate syntax, called 'local syntax', as 'natural' as possible to a human user. It also provides some features not available with the standard syntax.

Service Provided

The UCSB Computer Center operates an IBM 360/75 and runs OS MVT release 21.0 and HASP. All batch jobs at UCSB enter the system through HASP, and therefore RJS acts as an interface between the user and HASP. RJS's function is to provide the user with a HASP virtual RJE terminal, consisting of card reader, card punch, line printer and operator's console, and to manage the first three of these for the user in response to commands issued by him. By virtue of the fact that RJS maintains the correspondence between a particular user and the HASP RJE terminal owned by him, the user may issue commands to alter the status of those jobs submitted by him through his RJE terminal. This may be done even if the user has logged out of RJS and back in again, possibly from another site.

The sequence of events involved in using RJS are as follows. First, the user logs in, specifying a user name, password, and account number. In addition to indicating how subsequent use of RJS is to be billed, this accounting information identifies the owner of a particular RJE terminal. That is, the association between user name and HASP virtual RJE terminal is unique, and only one individual is allowed logged in under a given user name at a time.

At present, billing within RJS is not implemented, and therefore the login account number is completely arbitrary, and user name and password are relatively so. The first time a given user name is used, any password may be chosen; thereafter, as long as the user name is known to RJS, the same password must be used. A user name remains known to RJS while he is logged in, and when he is not, as long as he owns at least one job known to PJS. Otherwise, the user name is discarded.

After a user is logged in, he specifies input and output file information and requests input retrieval be initiated. He may then log out or not as he wishes; specifically he is not required to remain logged in during source file retrieval. A job can completed abnormally at any stage of processing, or normally, after storage of the last of its output. In any case, the circumstances of the final disposition of the job are displayed to the user immediately if he is logged in, saved for a period after its completion, and are available to him on request the next time he logs in. This status information is retained for at least a fixed period (currently two days), but will be retained longer as long as sufficient storage space is available for it.

RJS Commands - General Information

In order to simplify specification of job parameters, RJS maintains a set of accumulators for these parameters. Each accumulator is initially empty, and may have its contents set or referred to by various commands. The following parameter accumulators are maintained for each user (user name, password, and account together are termed accounting parameters): login accounting parameter (those specified either in the LOGIN or the USER, PASS, and ACCT commands), source pathname, print pathname, punch pathname, source accounting parameter, print accounting parameter, and punch accounting parameter. In addition, associated with each job are the parameters source, print, and punch pathname, and source, print, and punch accounting parameters.

When the TELNET connections are first opened, RJS sends the user a herald message of the form '300 UCSB RJS (VER. <date>) TTY <integer>.', where <date> identifies the current version of RJS, and <integer> identifies the user's terminal in the sense that each TELNET connection is assigned a unique TTY number.

During the process of running a job, any of several spontaneous job status messages may be displayed to the user. They are '240 INPUT RETRIEVAL FOR JOB <jobid> HAS BEGUN.', displayed when the input connection(s) have been established, '261 JOB <jobid> HAS COMPLETED EXECUTION.', when the first of the job's printed output has been received from HASP, '466 PRINTED [or PUNCHED] OUTPUT FROM UNKNOWN JOB (HASP JOB NUMBER <integer>) IS BEING DISCARDED.', if printed or punched output is received from HASP over the user's line for a job not known to RJS, and messages explaining errors such as ICP/RFC failure, invalid job card, null source deck, missing pathname, and data transfer network errors (see Appendix B for for a list of possible reply id's). In addition, while in local syntax, the user may receive messages from HASP over his virtual operator's console. Some are responses to HASP commands issued by the user, and others are spontaneous messages. All, however, are asynchronous to the entering of other RJS commands.

Certain general rules hold for both sets of syntax. They are as follows:

1. The character pair CR-LF is used as command accept to terminate commands. Other occurrences of these two characters are ignored, and may be used as desired for local carriage control.

2. Any number of blanks are permitted before or after any syntactic unit (including the first and last).

3. Upper and lower case alphabetics are treated identically everywhere except in <filename>.

4. Whenever a switch is made to the syntax not in effect, the current TELNET modes (full or halfduplex, character or line at a time) are saved, and reinstated when the current syntax is again entered.

In the syntactic descriptions below, the following notation is used:

'text'	denotes literal text (quotes not part of text)
<unit>	denotes name of syntactic unit
<a>_<b>	denotes choice of syntactic units <a> or <b>
[<a>]	denotes optional syntactic unit
(<a><b><c>)	denotes group of syntactic units treated as a whole
=	syntactic unit at left defined by statement on right

The following general syntactic definitions are hereby made:

<CA>	= command accept
<user name >	= a string of 1 to 8 alphameric characters
<password>	= a string of 1 to 8 alphameric characters
<account>	= a string of 1 to 6 alphameric characters
<jobid>	= a string of 1 to 8 alphameric characters, the first of which is alphabetic.

Description of RJS Commands, Standard Syntax

The following is a list of the commands supported in the standard syntax. Where not specified, the command's response is '200 OK'. In those cases where it is stated that the user must be logged in, the response '504 LOGIN PLEASE.' is displayed if the user is not logged in.

'USER' ['='] <user name> <CA>

Specifies the user's user name for accounting purposes, initiates login, and initializes the source, print, and punch user name accumulators to <user name>. To complete login, this command must be followed by a successful PASS command. The only other command allowed before the user is logged in is BYE. The response to a syntactically valid USER command is always '330 ENTER PASSWORD'

'PASS' ['='] (password> <CA>

Specifies the user's password to gain access to the user's account, completes the login initiated by a previous USER command, and initializes the source, print, and punch password accumulators to <password>. The response to a successful PASS command is '230 USER <user name> OWNS REMOTE TERMINAL <integer>.', where <integer> is the number of the HASP virtual RJE terminal owned by the user. The following error replies are possible: '431 NEW USERS ARE NOT BEING ACCEPTED AT THIS TIME.' if there is no free HASP terminal to assign to the user (there is currently a maximum of 10), or if there are insufficient other resources available to support another user; '431 INCORRECT PASSWORD.' if the specified <password> is not that assigned to the previously specified <user name>; '431 ANOTHER USER IS LOGGED IN AS <user name>.' - only one user is allowed logged in with a given <user name> at a time.

'ACCT' ['='] <account> <CA>

Specifies the UCSB Computer Center account number to which the user's use of RJS is to be billed, and initializes the source, print, and punch account accumulators to <account>. The response to a syntactically correct ACCT command is '200 OK'.

As previously stated, RJS billing is yet to be implemented, and therefore the ACCT command is optional. Users and writers of user processes are warned, however, that it will eventually be required, and that at that point, the PASS command will return a reply id of 330, and the ACCT command will have those replies listed above under PASS.

'BYE' <CA>

Logs the user out and closes the TELNET connections to him, regardless of whether there are any file transfers in progress; if there are, they continue unaltered. The response to a BYE Command is always '231 LOGOUT C0MPLETED; TRANSFERS (IF ANY) CONTINUE.', followed by a continuation line of 'TTY <integer> IS DISCONNECTED.'

'STATUS' <CA>

Lists the user names of those users currently known to RJS. The response is either '100 NO USERS ARE KNOWN TO RJS' or '10O THE FOLLOWING USERS ARE KNOWN TO RJS:', followed by one or more lines, each beginning with a continuation reply id of four blanks, giving a remote terminal number, the user name of the user who owns that terminal, and the name of the site from which he last logged in.

'SYNTAX' <CA>

Causes the current command syntax to become local syntax. The TELNET modes (full or half duplex, character or line at a time) most recently in effect in local syntax again become effective. RJS responds with the local syntax prompt character (currently number sign, '#').

The remaining commands require the user to be logged in.

'REINIT' <CA>

Resets to empty the source, print, and punch accounting parameter, the source, print, and punch pathname, and the login accounting parameter accumulators. The response to a REINIT command is always '204 OK'.

('INUSER' _ 'INID') ['='] <user name> <CA>

Sets the source user name accumulator to (user name>.

'INPASS' ['='] <password> <CA>

Sets the source password accumulator to <password>.

'INACCT' ['='] <account> <CA>

Sets the source account accumulator to <account>.

'OUTUSER' ['='] <user name> <CA>

Sets the print and punch user name accumulators to <user name>.

'OUTPASS' ['='] <password> <CA>

Sets the print and punch password accumulators to <password>.

'OUTACCT' <account> <CA>

Sets the print and punch account accumulators to <account>.

'INPATH' ['='] <pathname> <CA>
<pathname> = <file> _ <socket>
<socket> = [<host addr> ','] <socket number> <attributes>
<file> = <host addr> <attributes> '/' <filename>
<host addr> = <integer>
<socket number> = <integer>
<integer> = <decimal integer> _ ('D' <decimal integer>) _
('H' <hexadecimal integer>) _ ('X' <hexadecimal integer>) _
('O' <octal integer>)
<attributes> = ':T' _ ':A' _ ':N' _ ':' _ <null> _ ':E' _
':TE' _ ':AE' _ ':NE'
<filename> = 1 to 16 ASCII characters, excluding CR and LF
(codes 0 through 127, excluding 10 and 13)

Sets the source pathname accumulator to <pathname>. The <pathname> is the means for specifying a file's source or destination; its semantics are as follows:

1. Specification of <socket> indicates that RJS will establish a simplex connection to the stated socket (RJS issuing the CONNECT, user issuing a LISTEN). The data is then transferred over this connection, with CLOSE signaling end of file.

2. If <host addr> in <socket> is defaulted, the host containing the TELNET user will be assumed.

3. Specification of <file> indicates that RJS will contact the standard FTP server socket (currently socket 3) at the stated host, and transfer the data according to the File Transfer Protocol. The <file name> referred to here corresponds to the term <pathname> used in the FTP specification document, RFC #354.

4. The type of carriage control and the data code used are determined by <attributes>. The meaning of this parameter is as follows:

':T'	ASCII code, TELNET carriage control
':A'	ASCII code, ASA carriage control
':N'	ASCII code, no carriage control
':'	identical to ':N' for input, ':A' for output
<null>	identical to ':N' for input, ':A' for output
':E'	identical to ':NE' for input, ':AE' for output
':TE'	EBCDIC code, TELNET carriage control
':AE'	EBCDIC code, ASA carriage control
':NE'	EBCDIC code, no carriage control

Detailed descriptions of the transfer modes may be found below under 'RJS File Transfer.'

('OUTPATH' _ 'OUT') <output file> '=' <disp> <CA>
<output file> = 'A' _ 'B' _ <null>
<disp> = <pathname> _ '(H)' _ ('(S)' (pathname>) _ '(D)'

Stores <disp> in the print pathname accumulator if <output file> is either 'A' or <null>, or in the punch pathname accumulator if <output file> is 'B'. The meanings of the options for <disp> are as follows:

<pathname>	transmit-and-discard - the file is sent and then deleted
'(H)'	hold only - the file is not sent but rather held for user intervention
'(S)' <pathname>	save - the file is sent and then held for user intervention
'(D)'	discard - the file is deleted as soon as it is produced

'INPUT' <CA>

Creates a job, stores with it the contents of the source, print, and punch accounting parameter and pathname accumulators, and places it in a queue within RJS of jobs owned by the user awaiting source file transfer. When it becomes the first or only job in this queue, the retrieval of its source file is initiated. A job identifier ('jobid') is assigned to the job and displayed to the user. The contents of the source and print pathname accumulators must have been set by INPATH and OUTPATH commands previous to the INPUT command. If successful, the message '260 ASSIGNED JOBID IS <jobid>.' is

displayed, where <jobid> is that which may used in subsequent RJS commands to identify this particular job. If the user is found to own the maximum number of jobs (currently 5), an attempt is made to satisfy the request by finding the oldest of the user's jobs that has completed processing. If this can be done, the old job is deleted, and the response to the command is '260 JOB <old jobid> IS BEING DISCARDED TO MAKE ROOM FOR THE NEW JOB.', followed by a continuation line of 'ASSIGNED JOB ID IS <new jobid>.' The following error responses are possible: '360 SOURCE PATHNAME HAS NOT BEEN SPECIFIED.', '505 PRINT PATHNAME HAS NOT BEEN SPECIFIED.', '504 NEW JOBS ARE NOT BEING ACCEPTED AT THIS TIME.', and '504 USER (user name> ALREADY OWNS THE MAXIMUM NUMBER OF JOBS.'

'CHANGE' <jobid> <output file> '=' <disp> <CA>

Stores <disp> as the print pathname of job <jobid> if <output file> is either 'A', or <null>, or as the punch pathname if <output file> is 'B', if the appropriated file transfer has not yet begun. The following error replies are possible: '464 JOB <jobid> NOT FOUND.', '5O4 JOB <jobid> IS ALREADY BEING, OR HAS BEEN, PRINTED.', if <output file> is 'A', '504 JOB <jobid> IS ALREADY BEING OR HAS BEEN, PUNCHED.', if <output file> is 'B', and '464 USER <user name> DOES NOT OWN JOB <jobid>.'

'STATUS' <jobid> <CA>

Causes the status of the job known to RJS as <jobid> to be displayed. Included in this display are in which stage of RJS processing the job is ('BEING READ', 'IN EXECUTION', 'BEING PRINTED', 'BEING PUNCHED', or 'HAS COMPLETED'), the pathname information (accounting parameters, host name, socket number, attributes, disposition, and filename) for those files (source, print and punch) that have been supplied for the job, and if the job has failed at some stage of RJS processing, an explanation of the failure. The possible responses are '464 JOB <jobid> NOT FOUND.', and a line with reply code 161 followed by zero or more continuation lines explaining the status of the job.

'CANCEL' <jobid> <CA>

Causes processing of the job known to RJS as <jobid> to terminate immediately, and all record of it to be deleted. If FTP data and/or command connections are pending or established, they are closed; if the job is in execution, an OS CANCEL command is issued to terminate execution. Any output from the job is lost, and a subsequent request for status of job <jobid> will return the diagnostic that the job is not found. The successful responses are '262 JOB <jobid> DELETED.'

and '262 JOB <jobid> WILL BE DELETED AS SOON AS POSSIBLE.'; the possible failure responses are '464 JOB NOT FOUND.' and '464 USER <user name> DOES NOT OWN JOB <jobid>.'

The following standard RJS commands are as yet not implemented, and elicit the response '506 COMMAND NOT IMPLEMENTED.': ABORT, ALTER, BACK, HOLD, OP, RECOVER, RESTART, and SKIP.

RJS Commands - Local Syntax Conventions

In addition to those general conventions discussed above, the following rules hold for local syntax:

1. Except in certain circumstances, noted below, a period may be used for command accept.

2. The following control characters have the indicated functions:

SOH (control A)	delete last character
DEL	delete last character
SYN (control V)	delete last word
CAN (control X)	delete entire line
EOT (control D)	display current word
ACK (control F)	display entire line
'?'	display acceptable input forms
ESC	force recognition of current word
blank	force recognition of current word

3. The TELNET control characters 'you-echo' and 'I-echo' have the desired results. 'You-echo' also has the effect of changing to character at a time mode, if the user is not already in character at a time mode.

Description of RJS Commands, Local Syntax

The following is a list of those commands supported in local syntax. In those cases where no success reply is indicated, RJS responds with CR-LF followed by the prompt character. In those cases where it is stated that the user must be, or must not be, logged in, 'LOGIN PLEASE.' or 'LOGGED IN.', respectively, are displayed if the user is not in the appropriate state. When a reference is made to a response listed under a standard syntax command, it should be noted that reply id's are not displayed under local syntax.

'FULLDUPLEX' <CA>

Sets the user to fullduplex and character at a time modes.

'HALFDUPLEX' <CA>

Sets the user to halfduplex mode.

'LINE<-AT<-A<-TIME' <CA>

Set's the user to halfduplex, line at a time modes. The control characters previously described remain effective, but RJS will send no output over the TELNET connection except when the current command line is empty.

'LOGIN' <user name> <password> <account> <CA>

Specifies the UCSB Computer Center user name and account to which the user's use Of RJS is to be billed, logs the user in, and sets the source, print, and punch accounting parameter accumulators to <user name>, <password> and <account>. This command is valid only if the user is not logged in, and has the same replies as the standard syntax 'PASS' command.

'DISCONNECT' <CA>

Closes the TELNET connection. If the user is logged in, he is first logged out. The effective action taken in response to an unexpected close on the TELNET connection is that of a DISCONNECT. The response to a DISCONNECT command is 'TTY <integer> IS DISCONNECTED.'

The remaining commands require the user to be logged in.

'LOG0UT' <CA>

Logs the user out and terminates billing of subsequent activity over the TELNET connection to the previously effective accounting parameters, and performs the effective action of the REINITIALIZE command. LOGOUT does not close the TELNET connection, nor does it affect any file transfers in progress for jobs owned by the user.

'REINITIALIZE' <CA>

Resets to empty the following accumulators: source, print and punch accounting parameter, source, print and punch pathname, and login accounting parameter.

'ACCOUNTING' <account parms> <CA>
<account parms> = '(' <u> ',' <p> ',' <a> ')'
<u> = <user name> _ <null>
<p> = <password> _ <null>
<a> = <account> _ <null>

Sets the source, print and punch accounting parameters to <account parms>. Specification of <null> for any of <u>, <p>, or <a> indicates use of the contents of the corresponding login accumulator.

'SOURCE' <account parms> <CA>

Set the source accounting parameter accumulators to <account parms>.

'PRINT' <account parms> <CA>

Sets the print accounting parameter accumulators to <account parms>.

'PUNCH' <account parms> <CA>

Sets the punch accounting parameter accumulators to <account parms>.

'SOURCE' <jobid> (<account parms> _ <null>) <pathname> <CA>

Sets the source pathname of job <jobid> to <pathname>, and the source accounting parameters to <account parms>, if specified, or otherwise to the contents of the source accounting parameter accumulators. If job <jobid> already exists and its source pathname has not been specified, the new pathname is stored; if it has been specified, it is changed unless source file retrieval has already begun. If the job does not already exist, a new job is created and the pathname stored. Restrictions are that if a job with a given <jobid> has completed processing, it must be DELETE'd before that <jobid> may be used for a new job; a user may only alter jobs owned by him; and he may not own more than a certain fixed number of jobs (currently 5). If the user already owns the maximum number, an attempt is made to delete an old job to make room for the new one, as described for INPUT under standard syntax. The SOURCE command has the following possible error responses: 'NEW JOBS ARE NOT BEING ACCEPTED AT THIS TIME.', 'USER <user name> ALREADY OWNS THE MAXIMUM NUMBER OF JOBS.', 'USER <user name> DOES NOT OWN JOB <jobid>.', 'JOB <jobid> HAS ALREADY COMPLETED.', and 'JOB <jobid> IS ALREADY BEING, OR HAS BEEN, READ.'

'PRINT <jobid> (<account parms> _ <null>) <disp> <CA>

Sets the print pathname of job <jobid> to <disp>, and the print accounting parameters to <account parms> if specified, or otherwise to the contents of the print accounting parameter accumulators. The PRINT command either creates a new job or modifies an existing one, as explained under SOURCE, and has the same restrictions and error messages listed for the SOURCE command, after making the obvious substitution of 'PRINTED' for 'READ'. The PRINT command is valid only before print file transfer begins.

'PUNCH' <jobid> (<account parms> _ <null>) <disp> <CA>

Sets the punch pathname of job <jobid> to <disp>, and the punch accounting parameters to <account parms> if specified, or otherwise to the contents of the punch accounting parameter accumulators. The PUNCH command either creates a new job or modifies an existing one, like the SOURCE and PRINT commands, and has the same restrictions and error messages listed for the SOURCE command, after making the substitution of 'PUNCHED' for 'READ'. The PUNCH command is valid only before punch file transfer begins.

'DELETE' <jobid> <CA>

Identical in function to the CANCEL command in standard syntax.

'INPUT' <jobid> <CA>

Places the job identified by <jobid> in a queue within RJS of jobs owned by the users awaiting source file transfer. When it becomes the first or only job in this queue, the retrieval of the source file is initiated. If the INPUT command is successful, the message 'JOB <jobid> ACCEPTED FOR PROCESSING.' is displayed. The following error messages are possible: 'JOB <jobid> NOT FOUND.', 'USER <user name> DOES NOT OWN JOB <jobid>.', 'JOB <jobid> HAS ALREADY COMPLETED.', ' SOURCE PATHNAME HAS NOT BEEN SPECIFIED.' and 'PRINT PATHNAME HAS NOT BEEN SPECIFIED.'

'JOBSTAT' <jobid> <CA>

Identical in function and response to the 'STATUS' <jobid> command in standard syntax.

'JOBLIST' <CA>

Lists the jobid's of those jobs owned by the user.

$ <text> <CA>
<text> = a string of any characters including '?' and '.'.
Note that <CA> must be CR-LF, rather than period.

Issues <text> as a HASP operator command over the user's virtual operator's console. See Appendix A for a description of HASP commands and command responses.

RJS File Transfer

The <pathname> defined earlier is the means whereby the user specifies the location and attributes of his source, print and punch files. The means of determining a file's location have been previously discussed; this section explains the controls the user has over data attributes.

The parameter <attributes> specifies the type of carriage control and the mode of transfer. For the case of transfer over a simplex connection, this parameter has the following meanings:

':T' or ':TE' - TELNET-like carriage control. The data is a stream of characters with embedded carriage control bytes. Page eject is signaled by form feed, ASCII or EBCDIC decimal 12, new line by carriage return - line feed, ASCII l3-lO, EBCDIC 13-27. Multiple new line ('double spacing' or 'triple spacing') is indicated by multiple occurances of CR-LF.

':A' or ':AE' - ASA carriage control. The data is a series of fixed-length records, 81 characters on input, 133 on output, with the first character of each record an ASA carriage control character. The possible carriage control characters are as follows; '+' - no line advance before print (overprint), ' ' - one line advance (single space), '0' - two lines advance (double space), '-' - three lines advance (triple space), and '1' - page eject. Whatever carriage control character appears on input is ignored.

':N' or ':NE' - no carriage control. The data is a series of fixed length records, 80 characters on input, 132 on output. Any carriage control generated on output is discarded before transmission.

When file transfer takes place by means of FTP, the interpretation of the <attributes> parameter is somewhat different. In this case the meanings are as follows:

':T' or ':TE' - TELNET-like carriage control. The data has the same format as for the simplex connection, and is transferred in stream mode, file structure, and either ASCII ('A') or EBCDIC ('E') type.

':A' or ':AE' - ASA carriage control. Data is transferred in blocked mode, record structure, and either ASCII print ('P') or EBCDIC print ('F') type. The first character of every record is the ASA carriage control character described above.

’:N’ or ’:NE’ - no carriage control. Data is transferred in blocked mode, record structure, and either ASCII (’A’) or EBCDIC (’E’) type. As for the simplex connection, no carriage control information is present.

In order to effect the FTP file transfer, RJS issues the following FTP commands (in the given order): USER (if access user name has been specified), PASS (if password specified), ACCT (if account specified), BYTE (specifying bytesize of 8), ALLO (if outputting file), TYPE, STRU, MODE, SOCK, and APPE or RETR.

Appendix A: The HASP Spooling System

HASP is a spooling-queuing-scheduling system used in conjunction with IBM OS/360 to aid in processing of batch jobs. The main purpose of HASP is to increase throughput by minimizing I/O wait time and providing a priority scheduling scheme whereby shorter jobs are chosen for processing over longer jobs.

There are several stages of processing, or functions, within HASP. At any instant, a given job is either in some stage of processing, in which case the job is said to be active, or it is waiting to be processed by some function, in which case it is said to be queued for that function. Jobs to be processed by a function are selected from the queue of jobs waiting for that function, in order of decreasing priority. A job's priority is determined by its estimated CPU tine and volume of output. The result is that smaller jobs are selected for processing over larger jobs, and therefore spend less time in the system.

The HASP remote user is provided with a virtual operator's console. Over this console he may enter HASP operator commands to display information about the system in general, and to exercise control over his terminal and his jobs. HASP sends messages to his console in response to his commands, and to inform him of conditions concerning him as they arise. HASP commands have the following general form:

```
$ <verb> <operand1>,<operand2>,...,<operandn>
where
 <verb>          = a single character verb which
                   identifies the general function
                   to be performed

 <operand>       = identification of the object to
                   be displayed or acted upon.
```

Zero or more operands may be present, depending on the command, and commas are used to separate operands when more than one is used. In general, alphabetics may be entered in either upper or lower case, and for text outside paired apostrophes, blanks may be inserted at any point desired. Apostrophes intended as text characters must appear in duplicate.

Every HASP command ellicts one or more responses. The response "OK" is used in many cases to acknowledge the command and to signify that the requested action has been taken or initiated. In the later case, an information message will be issued when the request is completed.

Every HASP console message begins with the text 'S HH.MM.SS' or 'S*HH.MM.SS', where HH.MM.SS is the time of day in hours, minutes, and seconds, and in 24 hour clock.

Many commands display job status information as a response. The format of this standard response is as follows:

```
jobs queued for processing:
JOB jjj jobname AW EXEC  class PRIO prio HOLD
                   PRINT rem             PURGE
                   PUNCH rem             *DUP*
                   PURGE

jobs being processed:
JOB jjj jobname EXECUTING class PRIO prio HOLD
                ON DEVICE dev             PURGE
                IS PURGING
```

where

jjj	= HASP assigned job number
jobname	= OS jobname
AW	= 'AWAITING'
class	= job's job class
prio	= job's HASP internal priority
rem	= terminal number of remote terminal where job is queued to print or punch
dev	= device name
HOLD	= signifies job is in hold status, or will be at completion of current function
PURGE	= signifies job will be purged at completion of current function
DUP	= signifies job cannot begin execution until another job with same jobname completes

The following is a brief description of those HASP operator commands that may be issued by a remote user (for a more complete description, see NIC #16306):

SDA	Display status information on all active jobs
SDF [,rem]	Display number of jobs queued for special forms
SDN [,queue]	Display status information on all queued jobs
SDQ	Display number of queued jobs

SCJ jjj	Delete job immediately
SKJ jjj	Issue OS CANCEL and delete job immediately
SPJ jjj	Delete job after current function
SDJ jjj	Display job status information
SD'jobname	Display job status information
SB device,pages	Backspace device
SC device	Delete current function on device
SF device,pages	Forward space device
SDP device	Display job number of job on device
SDI	Display status and classes of initiators
SDLINE rem	Display status of remote terminal
SDRM rem	Display status of remote terminal
SDU	Display status of local unit record devices
SDM rem,'message'	Display message to remote console

Appendix B: RJS Reply ID's

The following is a list of the reply id's of the replies generated by RJS in response to the indicated commands:

command	success reply	failure replies
USER	330	501
PASS	230	501,431,505
ACCT	200	501
BYE	231	501
REINIT	2O4	5O1,5O4
INUSER/INID	200	501,504
INPASS	200	501,504
INACCT	200	501,504
OUTUSER	200	501,504
OUTPASS	200	501,504
OUTACCT	200	501,504
INPATH	200	501,504
OUTPATH/OUT	200	501,504
INPUT	260	501,360,504,505
CHANGE	200	501,464,504
STATUS (no operand)	100	501
STATUS (with operand)	161	501,464,504
CANCEL	262	501,464,504

Possible spontaneous reply id's are: 300, 440, 441, 442, 461, and 466.

[This RFC was put into machine readable form for entry]
[into the online RFC archives by Mikan Mirko]