

RESERV-MAC: Reserve Expandable System Enhancing
Record Verification, Management, and Circulation

Burling Library Reserve Book System

Author: Nathaniel Solomon Borenstein
Implementation Date: January 1978
Program Version 05
Last Documentation Update: June 5, 1978

This document describes the current version of the RESERV system for Burling Library. Any suggestions regarding modification of either the RESERV system or this explanatory document are welcomed. Any questions or suggestions stemming from the language of this document or the functioning of the RESERV system should be directed to the author.

NOTE: This manual will be updated periodically as modifications in the RESERV system are made. The most current version will be available at the reserve desk on two hour reserve.

CONTENTS:

- I. Introductory Remarks: The RESERV system, its purpose and operation. How to type on a computer keyboard. Error procedures.
- II. Getting started: How to log onto the computer and run the RESERV system.
- III. Publicly accessible features: How to find out information about reserve readings, end the program, or log out. (These features may be used by anyone logged into the computer.)
- IV. Everyday Library-only features: How to check books and keys in and out, record payment of fines, etc. (These features may be used by anyone logged into the library account.)
- V. 'Privileged' Librarian-only features: How to add, change, and delete entries to the reserve files, change passwords, request special printouts, etc. (These features may be used only by someone logged into the library account who knows the required password.)
- VI. Special Override procedures: A few simple ways to show the machine who is boss. (You.)
- VII. Technical Overview: Chaining structure, command parsing, security considerations, required support software, etc.
- VIII. Data Structure: Data storage structure and the storage/retrieval functions.
- IX. Programming aids: Hidden options and the DOCTOR program.
- X. Appendix: Summary of RESERV-MAC commands.
- XI. Index.

Some of the above-mentioned sections of this manual may not be present in every copy of this document.

I. Introduction

RESERV is a set of computer programs written for the use of the staff of Burling Library to streamline the operation of the reserve book system. The computer keeps information on file about each item that has been placed on reserve; specifically, it keeps a record of the title, author, and coding of each book on reserve, and keeps analogous information (dates for periodicals, pamphlet #'s for pamphlets, etc.) for other reserves. It also keeps information about the length of time for which the item can be checked out, number of copies, etc. One special feature of the RESERV system not found in most systems of its kind is 'dynamic string storage'. This means that there are no limitations placed on the length of the titles, authors' names, and coding information for each reserve item.

All of the information is stored on a magnetic disk in the computer center, and information is added, deleted, or altered by the programs of the RESERV system according to a simple series of commands. The principal benefit of the RESERV system is that it saves time for library staff members by automatically printing overdue notices daily (except weekends), fine notices twice weekly, a monthly statistical summary of reserve book usage, library cards upon request, updated catalog pages whenever books are put on reserve, and additional cards and catalogs whenever requested. Also, because the RESERV system can identify students by ID numbers, there is a decrease in the risk of books being checked out under false names and perhaps never returned.

Typing on a Computer Keyboard

Those who have never used the computer before will be pleasantly surprised by three useful features that are not available on a regular typewriter keyboard. The first is the <DELETE> key, which simply deletes the previous letter. (Try it-- soon you'll wish typewriters had <DELETE> keys.) The second is the <CTRL/U> combination, made by depressing the <CTRL> key and typing 'U'. This deletes the entire line that has been typed since you last hit the <RETURN> key or the <LINE FEED> key. Last (but far from least) is the <CTRL/C>, which aborts the program currently being run. <CTRL/C> is what you should type if you or the computer are doing something you do not want to do.

Often the computer will ask you a yes-or-no question. Whenever this occurs, any answer beginning with a 'Y', such as 'YEECH' will be treated as a 'YES', while any answer beginning with an 'N' is 'NO'. An answer that does not begin with either a 'Y' or an 'N' will probably be treated as an 'N', but it is safest to answer 'Y' or 'N'. If a question

requires anything more than a simple yes or no answer, you can usually get help by typing a question mark ('?') instead of answering.

Error Procedures

Although computers never make mistakes in the sense that they always do exactly as they have been told, nonetheless the propensity of human beings (programmers included) to speak imprecisely is the bane of a computer's world. For that reason it seems inevitable that, on occasion, the computer will shout 'UNANTICIPATED ERROR', and spew out a few lines of gobbledygook ending up with 'PROGRAM TERMINATED'. If the error prevents you from checking books in or out you can, of course, always go back to manual checkout. In any event, however, it is a good idea for you to write down exactly what you were trying to do, what you said, and what the computer said when it freaked out. This will make it easier to find and correct the error.

In addition to programming errors, you may also have to deal with system shutdown. The computer is shut down regularly for routine maintenance-type work, and it may also simply die unexpectedly on occasion. Naturally, the computer center staff seeks to minimize the frequency of these occurrences, but they do happen now and then. When the computer is down, books may be checked out manually and then simply checked in manually when they return. However, if a book was checked out on the computer but is returned while the computer is down, it may be easy to forget to tell the computer that the book has been returned. If you don't tell it, of course, it won't know, and it will send out overdue notices that are completely unwarranted, simply because it won't know any better. Thus if a book is returned while the machine is down, the computer should be told of the return as soon as it starts working again.

It is also worth mentioning that the RESERV system is designed to notice and correct certain internal errors which may occur in the checkin-checkout process. These will not affect the user at all, and in fact probably will never even occur. A record of all these errors as well as of any errors that may occur during a batch job run can be found in the file ERRLOG.DAT.

II. Getting Started

First, turn the terminal on. On most terminals, the switch is on the right side. It may take a few seconds to warm up, like a TV does. Then prove that you are human by typing:

HELLO

and hitting the <RETURN> key. (Always hit the <RETURN> key after you say something to the computer; that's the only way it can tell you've finished saying it.) The computer will respond with a line of extraneous gibberish similar to:

RSTS V06B-02 Grinnell Coll. Job 13 KB41 09-Feb-79 09:17
#

which means, roughly, "Who are you?" So, tell it who you are:

11,10

(11,10 is the library account number.) The computer will remain skeptical and ask you to prove it with a password:

PASSWORD:

You type in the library password. Note that at this point, as you type, the letters being typed will not appear on the screen. This is the computer's way of protecting your secret word from public eyes. If you don't know the password, you're out of luck. If you give it the right password, it will start being friendly, give you any special messages it might have for you, and say:

READY

This means that it will do anything that you tell it to do, within the limits of its ability. Since you want it to run the RESERV program, you say:

RUN (5,10)RESERV

(The RESERV program is stored in account 5,10.) It will then bid you a good morning, print any special messages it may have for users of the RESERV system, and then indicate its readiness for RESERV commands by typing '#'. The number sign ('#') is the signal that the RESERV system is running and ready for your commands. The rest of this manual will explain those commands. Remember that if you are ever unsure how to respond to something the computer says, you can probably get help by typing a question mark ('?').

III. Publicly accessible commands

Most of the functions of the RESERV system can naturally only be executed by librarians; clearly it would not do to allow just anyone to record a book as being returned. However, there are a few functions that are simple and harmless and may be run by anyone, no matter what account he is in. These are the 'HELP' commands. As usual, you start by typing:

```
RUN (5,10)RESERV
```

The system will respond with a header line, and will then print the number-sign ('#') to show that it is ready for a command. If you are not logged into the library account, however, it will accept only a few commands. The simplest command is just a question mark ('?') or the word 'HELP'. This will cause the machine to explain to you exactly what you are allowed to do, and then let you choose one of the options it gives you. There are eight such options, and four of them are self-explanatory; they simply list the types of commands that are acceptable in various contexts. These may not do you too much good because you won't be allowed to use most of them unless you're logged into the library account. The other four help options, however, may be less futile. One is a system 'STATUS' report, listing all items currently checked out. (This report may be generated directly by typing 'STA' or 'STATUS' in response to '#'.) Another thing you can do is get a list of all items on reserve for a certain professor, department, or course. Simply type a question mark followed by a fragment of the professor, department, or course name. For example, typing '?MATH' will produce a listing of everything on reserve for the Mathematics department. Notice also that typing '?ATH' would produce, in addition to a listing of all reserves for the Math department, a listing of all items on reserve for Professor Atherton of the Art department and Professor Rather of the Music department. And '?E' would produce a listing of all reserves for any professor or department with an 'E' in it.

Another feature available to all users of the RESERV system allows the user to find out all the information on file about a certain reserve reading. This assumes that you know the reserve code (identifying number) of the reserve reading. If you don't know the reserve code, but do know the professor or department under which it is listed, you can find out the reserve code by typing a question mark and the professor or department name as described above. Anyway, once you have the reserve code (for example, 30047), simply type a question mark followed by the reserve code (e.g. '?30047'). The computer will then list all the

relevant information about the reading, including who it is now checked out to, the time due, and the number of people on the waiting list.

Another publicly available feature is somewhat trivial; if you type '?PROF' or '?P?' in response to '#', the machine will list all professors, departments, and courses that now have books on reserve. This may be useful in identifying a reserve reading if you don't even know which professor it might be listed under.

To put all of these together, imagine that you know that a book is on reserve but have no idea how to find out about it. First you could type '?PROF' to get a list of all professors with items on reserve. Then you could choose a likely professor or department and type a question mark followed by his (or its) name. If the book you are looking for is not in the list of books the machine gives you, you can try again with another professor or department. If it is on the list, you can note the reserve code that identifies it, and then get a full report on the book by typing a question mark followed by the reserve code.

Finally, you can search a professor's list selectively by typing, for example, '?MATH/STAT', which will search through the entries for all professor/course listings containing 'MATH' for any reserves that have the phrase 'STAT' somewhere in the author or title. You can also use a wildcard (e.g. '?*/STAT') to search through all reserves for the occurrence of a certain phrase in any title or author, but this can only be done from the library account and requires a password, and anyway is not advisable because it is so slow. A faster wildcard search is anticipated in future versions of RESERV-MAC.

Getting out: The 'LOGOUT' and 'END' commands

Use of the RESERV system may be terminated in one of two ways. Typing 'E' or 'END' in response to '#' causes the use of the RESERV system to terminate, and the user is returned to immediate mode of RSTS/E, the READY state. Typing 'LOG' or 'LOGOUT' in response to '#' automatically logs you off the computer.

IV. Everyday commands

This section will explain the RESERV commands used for checking things in and out. To check a reserve item in or out, simply type the reserve code in response to '#'. For example, typing '3004' will tell the computer that you want to check reserve #3004 in or out. (If there is no such reserve it will politely tell you so.) If the book is currently in the library, it will ask you for the name or ID# of the student who is checking it out. If the person checking the reserve out is a professor, you might wish to type 'PROF ' or 'PROF.' before his name. This will tell the machine that he is a professor and thus it will refrain from sending him overdue notices. NOTE: A quicker way to check out a book is to enter all the relevant information at once, separated by a comma or slash. For example, typing '1001,John Doe' or '1001/John Doe' will check out book #1001 to John Doe.

It might be noted here that if there is a waiting list for the item you are trying to check out, the computer will not allow anyone other than the appropriate person on the waiting list to check it out. (Of course, a book can only be held for fifteen minutes for each person on the list.) It is possible for you to override the waiting list at this point, so that the machine will ignore the list and check out the book to whomever you name. This is described in the section on 'Special Override Procedures'.

If you enter the reserve code of a book that is currently checked out, the computer will give you a choice of what you want to do about it. It will print out the title of the reserve, and will ask you to choose from the options displayed. ('N' for renewal, 'R' for return, 'W' for going on a waiting list, etc.)

Incidentally, a quicker way to accomplish the same thing is to just type 'R1001' or 'N1001' or 'W1001' instead of typing '1001' and then choosing 'R' or 'N' or 'W'. But either way will work equally well. If you have chosen to return the item ('R'), the computer will mark the item down as having been returned. (If there is a waiting list, it will tell you to hold the book for the first person on the list.) When finished, it will return to '#' to await your next command. If you choose the renewal option ('N'), the computer will renew the book and tell you when it is due. It will not renew the book if there is a waiting list, but it will tell you that the book must be returned promptly. If you choose the waiting list option ('W'), the computer will ask you for the name or ID# of the person to be placed on the list. It will then place that person on the waiting list for the appropriate reserve item. NOTE: A faster way to place an individual on a waiting list is to combine the

entire command in one line by typing 'W1001,John Doe'. This will put John Doe on the waiting list for reserve #1001.

It is also worth mentioning that if a book is renewed or returned after it was due, the computer will tell you what the fine is and ask you if it is being paid now. If it is not being paid now, the computer will keep a record of the transaction and will send the student a notice twice a week reminding him that he owes the library money. It is also possible to override the computer when it tells you there is a fine (aside from the obvious tactic of simply telling the computer that the fine is being paid). This is very simple and is described in the section on 'Special Override Features' later in this manual.

KEYS

Keys are checked out in much the same manner as books, except that the renewal and waiting list options obviously do not apply. To check key #3 in or out, just type 'K3' in response to '#', and everything will proceed as it does for reserves. If you just type 'K', the computer will list the keys by number and ask you which one you want. Once again, it is possible to conduct the entire transaction on one line by typing 'K3, John Doe' to check key # 3 out to John Doe. Incidentally, the computer will know which students are authorized to check out key #10 (the reading lab), and will not allow unauthorized students to check it out. This feature, too, can be overridden to allow anyone at all to check out the key to the reading lab; see the section on 'Special Override Features' for details.

Paying Outstanding Fines

Finally, if a student wants to pay a fine for which he has been receiving nagging computerized notices in the mail, just type '\$' in response to '#'. The computer will then ask you for the student's name or ID#, will tell you how much money he owes, and will ask you if he is paying it. If you answer 'Y' or 'YES', the student will stop receiving the notices. Note that you still have to write down the transaction in the money book at the circulation desk.

Incidentally, after you type '\$' and the computer asks you for the student's name or ID#, you can get a list of all students who currently have fines outstanding by typing '?' instead of a student's name. NOTE: A faster way to pay a fine for a person is to type the command all on one line by simply entering a dollar-sign followed by his name-- e.g. entering '\$JOHN DOE' rather than first entering '\$' and then entering 'JOHN DOE'. In this way you can also get a list of outstanding fines with the one line command, '\$?'.

V. Privileged Commands -- for Librarians only

The following commands (in response to '#') require the entry of a password and are intended only for the use of library staff members:

- 'ON'-- Puts books on reserve. This command will be explained in detail later.
- 'OFF'-- Takes books off reserve. This command will be explained in detail later.
- 'CHA' or 'CHANGE'-- Corrects erroneous information about reserve readings. You will provide a reserve code, and the computer will list all information about that reserve and ask you which information is incorrect. It will then ask you for the correct information and will have new cards and catalog pages printed out later for all copies of the item corrected. Note that you can shorten this process by typing 'CHA 1001' in response to '#' instead of first typing 'CHA' and then typing '1001'. Also, you can change a misspelled professor or department name by typing 'CHA *' in response to '#', or by typing 'CHA' and then entering '*' as the item to be corrected. You will then be asked to give, first the old and then the new spelling of the professor/course entry.
- 'CAT' or 'CATALOG'-- Requests that a new edition of the reserve catalog be prepared. You will be given seven options: First, you can request that the entire professorial catalog be reprinted (you specify the number of copies). Second, you can request that catalog pages for a certain professor/course listing be reprinted (again, you specify the number of copies). Third, you can have the machine go down the list of professor/course entries, asking you how many copies you want of each professor's listing. The last four options request the printing of overall author catalogs, overall title catalogs, overall dual author/title catalogs, and overall shelf lists. In each of these cases, once you choose the option the computer will simply ask you how many copies of itn effocfoaein beiedcv kcp zeriø erx zadd bephn ai ic ln fonfeonx deino eh hfnbasanxW Znbepn cs itn zek itah ah xcrnø ai ah onbbc"nrxnx itei as kcp onupnhi crn cs itnhn scpo ikfnh cs beiedcvhø itnr kcp htcpdx zeai e snz tcpoh lnscon onupnhiarv ercitnoW Ni zadd zcoq edd oavti as kcp onupnhi itn" ei effocga"eindk itn he"n ia"nø lpi itnk zadd ln savtiarv nebt citno sco xahq ebbnhh erx itph zadd rci opr eh nssabanridk eh as itnk znon oprrarv edernW

All catalogs will print out automatically as requested at the computer center. In addition, an extra copy of the entire catalog will be produced automatically on the first day of each month, and this copy will include circulation statistics--the number of times each book has been checked out, etc.

'CAR' or 'CARDS'-- Requests that a new card be prepared for an item currently on reserve. Since cards are prepared automatically when items are first placed on reserve, and when they are corrected with the CHANGE command, this is only necessary if an old card is lost, ruined, or used up. The computer will ask you for the reserve code of the item for which you want a card, and will then ask if you have any more requests for cards. No cards will actually be printed until the 'CARDPRINT' command is given. If you simply type 'CARDPRINT' in response to '#', this will cause the computer to print out all the cards that are waiting to be printed.

'MES' or 'MESSAGE'-- allows you to replace the text of the message printed out on overdue notices with a new one. It will simply ask you to type in the new message. The message you give will then be printed on every overdue notice sent out.

'PAS' or 'PASSWORD'-- allows you to change the library password. The computer will simply tell you what the old password was, and ask you what the new one should be. If the word you enter is a valid password, it will change the password used for all library programs and for logging in to the library account to the word indicated. Please note that if you use the system program for changing passwords ('\$CHANGE'), it will change only the LOGIN password and not the RESERV password. Thus the best way to change the password for the library account is to use the 'PAS' command in response to '#'.

'REA' or 'READING LAB' -- Allows you to modify the list of people authorized to use the reading lab. First it will ask you if you want to edit the list of authorized users of the reading lab. If you answer affirmatively, it will then list the authorized reading lab users, one by one, and allow you to judge whether or not to leave that person on the list. The name will remain on the list, unchanged, if you just hit <RETURN>. If you type 'K' (short for 'KILL'), the name will be deleted. If you type 'C' ('CHANGE'), the computer will allow you to enter another name in place of the old one. When you have finished this editing (or if you chose not to edit in the first place),

you will be allowed to enter the names or ID#'s of students whom you want to authorize to use the reading lab.

'ADD'-- allows you to add new students to the roll of students. You simply enter the new student's name.

Putting books on reserve: the 'ON' command

When you want to place a book or other item on reserve, you type 'ON' in response to "#". After the entry of a correct password, the computer will begin asking a series of simple questions about the reserve reading. The first question is simply:

NAME OR NUMBER OF PROFESSOR/COURSE ('?' FOR A LIST)?

You respond by entering the name or identifying number of the professor who is putting the item on reserve. (NOTE: The professor's name and all other textual material should be entered exactly the same way that it is to be alphabetized. Thus it should be entered last name first. Commas may be included if desired. If a professor's name is inadvertently entered as "JOHN SMITH (ECONOMICS 101)" instead of "SMITH, JOHN (ECONOMICS 101)", it will be alphabetized in the card catalog under "J" instead of "S".)

It is worth noting at this point that you can save a little time by typing 'ON SMITH, JOHN (ECONOMICS 101)' instead of just 'ON' in response to '#'. This will cause the first step of the ON program to be omitted, assuming that "SMITH, JOHN (ECONOMICS 101)" really is a valid professor/course entry. You can also omit the latter part of the professor's name; the last name alone should be enough, and the computer will print out the entire entry that it believes you are referring to. Note, however, that if there are two professors named 'SMITH', and you simply enter 'SMITH' instead of the full name, the computer will choose the first 'SMITH' alphabetically. Thus if two professors have the same last name, you must use the full name if you want to be certain of identifying them correctly.

Typing a question mark ('?') instead of a name will generate a listing of those professors who currently have books on reserve, identified by name and number. You may then enter either the name or the number to identify the professor.

The entry of the professor's name causes the program to look for a file of reserve books for that professor. If no such file exists, the computer will tell you that the professor has no books on reserve, and will ask you if you wish to create a reserve file for him. You may then respond with either "Y" or "N". If you answer 'Y', the computer will ask you for the name of the professor, department, and course (although the latter is optional and you can indicate that there is no specific course number by simply hitting <RETURN> instead of answering), and then continue as it would have done if a file had already existed for that professor. It asks:

TYPE OF MATERIAL (TYPE '?' FOR HELP)?

to which you respond 'BOOK', 'PC', or something similar. (The options are listed in response to '?'). If you respond 'XLIST' or 'OLD', this means that you want to crosslist an old reserve for another professor. The computer will simply ask what the old reserve number is, and will then crosslist the old reserve for the previously specified professor. If any type of material other than 'OLD' or 'XLIST' is specified, then the computer will ask for the title, author, number of copies, etc. Finally the computer will ask:

TYPE OF RESERVE (0=2 HR, 1=1 DAY, 2=3 DAY)?

to which you must enter either 0, 1, or 2, indicating the length of time for which the reserve material may be checked out. At this point the computer will recite all the information it has thus far been given and ask you to confirm that it is correct. If you answer negatively ('N' or 'NO'), the material will NOT be placed on reserve. If you answer positively ('Y'), the computer will then assign the reserve number, which will be printed on the reserve's card and which will be used to identify the reserve in all later dealings with the computer.

One unusual and very useful feature of the ON program is the use it makes of the <ESC> (ESCAPE) key. At almost any time, while putting an item on reserve, hitting the <ESC> key will allow you to go back to the previous question. This is very useful because it means you can correct typing errors without re-entering everything. Naturally you can go back two questions by hitting the <ESC> key twice, and so on.

Once the computer has finished putting the new item on reserve, it will ask you ('Y' or 'N') if you want to put another item on reserve for the same professor. If you do, it will start over with the 'TYPE OF MATERIAL?' question. Otherwise, it will ask you if you want to put another item on reserve for a different professor. If you do, it will

start over with 'PROFESSOR'S NAME OR NUMBER?'. Otherwise, it will return to '#' and await a new command.

Finally, when the title of a new reserve is entered, the computer will automatically append any leading articles to the end of the title. Thus 'The Way of Zen' becomes 'Way of Zen, The'. However, it should be noted that when the title is changed via the CHANGE command, no such alteration will take place. This is designed to allow you to change titles literally if you so desire, but you should keep in mind that unlike the ON program, the CHANGE program will alphabetize titles exactly as they are given. A similar situation exists with regard to the volume number for periodicals; if you enter the number alone in the ON program, e.g. '64', the computer will automatically append it to the phrase 'vol. ', e.g. 'vol. 64'. Again, however, the CHANGE program will enter the information literally as you give it.

Taking books off reserve:
The 'OFF' command

To take books off reserve, you enter 'OFF' in response to '#'. Again, a password is required. The computer will then ask:

ENTER THE RESERVE CODE OF THE ITEM TO BE TAKEN OFF RESERVE:
RESERVE CODE?

You may then enter the reserve code of any existing reserve. The specified item will then be removed from the list of reserves. NOTE: Books may be taken off reserve via a single command line. Simply type 'OFF 1001' instead of first typing 'OFF' and then typing '1001'.

As a precaution, whenever books are taken off reserve the computer will ask you to confirm that you have not made a typing error, and that you do indeed want to take the specified item off reserve. Also, books that are currently checked out may not be taken off reserve.

Finally, note that if you enter an asterisk ('*') as the entry to be removed, this means that it is the end of the semester and you want to remove a whole bunch of items from the reserve book list. This can be done only when there is no other person using the RESERV-MAC system. It will go through each professor, one at a time, asking you whether or not you wish to save his reserves for next semester. If you answer positively, the reserves for that professor will be saved. If you answer negatively, the reserves for that professor will be taken off reserve. A catalog of all items taken off reserve at this time will be

prepared in shelf-list form to help with the actual physical removal of items from the reserve shelves. Once the computer has gone all the way through the list of professors, it will sit there for a while and think. (Actually it will be processing the mammoth file rewrite demanded by the semester's end.) It will give a countdown to give you some idea of how long you will have to wait. When it is done, the RESERV-MAC system will be operational for the next semester, and a shelf-list of books taken off reserve will soon arrive in the campus mail. Of course, you can then get a shelf-list of books that remain on reserve for purposes of cross-checking the reserve shelves. At this point, you should call the computer center and tell them that you want the files '*.OLD' and '*.CAT' saved on a magnetic tape. These files are simply the data files for the RESERV-MAC system before the semester ending program was run. Having the files saved like this may prove a handy reference in future years. The computer center can delete the '*.OLD' files from disk once they have been saved on tape, but the '*.CAT' files must be preserved on disk for the next semester.

Note that you are not allowed to use CTRL/C to abort the end of semester program once it has started. However, once the computer has gone through the entire list of professors' names, it will give you one last chance to abort the program by asking you if you want to abort. If you choose to abort at this point, then the computer will spend a few moments fixing up the data files and return you to the number-sign ('#'). If you do not choose to abort at this point, then the aforementioned long processing period will begin, and no abort will be possible.

VI. Special Override Procedures

There are certain things that can be done to circumvent various safeguards against abuse of the RESERV system. These override procedures exist because they are very convenient in some circumstances; yet because they override certain safeguards, it may be desirable to limit the number of people who are aware of them.

The first special override involves the waiting list. When a book is being held for someone on a waiting list, any attempt to check it out yields the message:

THIS BOOK IS BEING HELD FOR JOHN DOE -- IS THAT WHO IS CHECKING IT OUT?

If you answer yes, the computer checks the book out to John Doe. If you answer no, it won't check it out to anybody and returns to '#'. However, if you simply hit the <ESC> (ESCAPE) key, the waiting list is overridden and destroyed. Thus a sample dialogue might go as follows: (underlined words are the ones you enter; non-underlined words are those of the computer.)

```
# 1001,JOHN DOE
THIS BOOK IS BEING HELD FOR RICHARD ROE -- IS THAT WHO IS
CHECKING IT OUT?  <ESC> (user hits <ESC> key)
OK,  JOHN  DOE  IS CHECKING OUT 'A PRIMER OF OPERANT
CONDITIONING'
DATE DUE:  24-MAY-86
TIME DUE:  5:32 PM
```

A similar override exists when a book is returned or renewed late; the computer says that there is a fine, and asks if it is being paid now. The user can simply hit the <ESC> key to cancel the fine entirely without paying it.

Finally, an override exists to allow a non-authorized person to check out the key to the reading lab. Simply type 'K10' in response to '#'. (Key # 10 is the Reading Lab.) The computer will ask for a name or ID#. Just enter the name or ID# as you usually would, but instead of hitting the <RETURN> key when you have finished typing the name, hit the <ESC> key. This will tell the computer to allow the name you entered to check out the key even if he is not on the list of authorized names.

Other Special Features

There are a few other special features of the RESERV system operating on RSTS/E that are worth noting. First of all, the statistics regarding how many books are checked out daily are kept in a file called 'DAILY.DAT'. If you are logged in to the library account, you can look at this file by typing 'PIP DAILY.DAT'. The daily records will then print out at your terminal. To have the daily records produced on paper, type 'QUE DAILY.DAT'. This will cause the daily records to be printed on the high-speed line printer at the computer center. Note that these are RSTS/E commands, not RESERV commands. That means that they are meaningful only when typed in response to 'READY', not in response to '#'. The summary of overdue notices sent out this semester is kept in the file 'NOTES.SUM', and may be similarly accessed via the commands 'PIP NOTES.SUM' and 'QUE NOTES.SUM'. These files are printed out automatically at the end of each semester, after which they are renamed with '.OLD' extensions and saved on magnetic tape. For more information on the PIP and QUE commands, see the RSTS/E System User's Guide, copies of which are available at the computer center as well as the Reserve desk of the library.

Also, the most recent four sets of card labels that have been printed out by the 'CARDPRINT' command are saved on disk so that they may be reprinted if necessary. Thus if the cards for one printout are somehow lost or destroyed, they may be salvaged using the QUE command with the special forms option. Cards for the last four printouts are kept in the files 'LATEST.CRD', 'BACKUP.CRD', 'OLD.CRD' and 'OLDEST.CRD'. You can look at them to figure out which you want by saying 'PIP OLDEST.CRD', etc. Then you can have the appropriate cards printed out on labels by saying 'QUE LIBRAR/FORMS:LABELS=OLDEST.CRD'. This tells the computer not merely to print out the file, but to print it on the appropriate type of form-- in this case, labels with three labels across and six lines per inch.

Another special procedure involves the daily batch jobs that run for the RESERV system. Each day two batch jobs run, one to print overdue notices and one to print cards and catalog pages. Each batch job queues the batch job to follow it, so that they can continue indefinitely without ceasing, even when school is not in session. These batch jobs can be stopped by typing 'STOP' in response to '#'. They can be restarted by typing 'AUTO'. This will cause the catalog pages that have been waiting for the batch jobs to be restarted to print out shortly, and will set up the regular batch structure to begin again with overdue notices the following morning. WARNING: If you queue the batch job to run using the 'AUTO' command when there is already a library batch job waiting to run, you will cause each batch

job to be executed twice daily indefinitely. This is wasteful and will also result in duplicate copies of all overdue notices (although probably not of cards and catalog pages). Thus you should never queue a batch job unless you are quite sure that it has not yet been queued. For a list of all batch jobs that have been queued, type 'QUE/L BA:' in response to 'READY'. For more details, see the RSTS/E System User's Guide.

Finally, if at the beginning of the semester you realize that you have a whole new list of authorized names for the reading lab and you just want to discard all the old ones, there is an easier way to do it than using the 'REA' command to delete each name individually. You can just say "KILL 'REALAB.DAT'" in response to READY. This tells the computer to delete the file 'REALAB.DAT' from the disk. Never do this with any other file names, unless you are certain you know what you are doing. You could destroy all of the library's data records in this manner, if the 'KILL' command is used indiscriminately.

VII. Technical Overview

RESERV is a system of nineteen programs written in BASIC-PLUS (NO EXTEND mode) for implementation on a PDP-11 running under RSTS/E, Version 6B. Several of the programs make use of certain cursor control characteristics of the CRT's on which they were designed to run, VT50's. RESERV will run with no operating errors on a terminal that does not have VT50 compatible cursor controls, but the result may not always be aesthetically pleasing, and a few excess letters may print out where cursor controls belong. All VT50 cursor controls begin with an ESCAPE character (CHR\$(155)), so the programs can be modified for other terminals by simply searching for all occurrences of 'CHR\$(155)' in the source programs.

RESERV may also be implemented under other versions of RSTS and perhaps (albeit with difficulty) under other systems supporting an equally powerful BASIC, but attention must be paid to the SYS calls that occur in the programs.

Also, RESERV assumes the presence of the RSTS/E program '\$QUE', to which it chains to print catalogs, library cards, and overdue notices. Use is also made in the CARDS program of the special forms option of the \$QUE program, which is installation-dependent but easily modified.

RESERV-MAC also requires the presence of the RSTS/E program '\$SORT', which is used in the alphabetization of shelf-lists and overall catalogs by the CATTLE and ENDSEM programs.

In addition, RESERV requires batch spooling facilities to run the daily batch job; the batch structure might have to be altered somewhat to run on another operating system.

Finally, one small section of the RESERV system deals with changing passwords. It solicits a new password, and changes both the internal RESERV password and the LOGIN password to match it. To do this, it chains to the program '\$CHANGE', which is Grinnell's password changing program. The four lines of the MORDTL program which deal with this must be altered for another implementation.

Security Considerations

It is recommended that there be two accounts for the RESERV system, one in which to store the data files and one from which to run the program. This will help to prevent

accidental destruction of files. Since the RESERV system does not currently run with privileged status, this requires that the run account have write access to the data files in the storage account. For maximum security, it is thus desirable to give these two accounts the same project number and different programmer numbers, and to use this project number exclusively for library accounts. The data files can then be given a protection code that allows read access to everyone but write access only to members of the same group (users with the same project number). Each program starts by assigning values to the variables A1\$ and A2\$. A1\$ is intended to be the account in which the files are stored; it is used in OPEN and CHAIN statements. The files to be stored in the A1\$ account should thus be all the source files (although the .BAS files should be stored and accessed via &LIB), as well as the compiled files (compiled at <104>-- they themselves will check the PPN of the job running to decide what is and is not permissible) and the primary data files (LIB.DAT, PERM.DAT, CIRC.DAT, STUDES.DAT, CARDS.DAT, and REALAB.DAT). Other files, of a more temporary nature, will be created as needed on the account in which RESERV is being run. A2\$ is the authorized run account; RESERV and the privileged programs will check the PPN of the job running against the PPN stored in A2\$, and will not allow any privileged commands to be carried out unless the two PPN's are identical. Privileged programs perform this check before doing anything. RESERV itself performs this check while analyzing the command string, as described below.

It should be noted that certain temporary files, such as those created for waiting lists, are automatically created and destroyed in the run account, and not the storage account. It might also be noted that if the run and storage accounts are the same account, as might be deemed desirable, then the above discussion of which files are located where is quite moot.

Command Parsing

The RESERV program is the command scanner and interpreter for the RESERV system. When accessed via the RUN command it prints a header line identifying the version running and the operating system name (error message # zero), and then checks an internal designator to see if this is the first time it has been run today. If so, it kills all existing waiting lists and prints out circulation statistics for the previous day. When chained to by another program in the system (or when run from any account other than the library run account), it skips those steps. Next it opens all files ('OPEN FOR INPUT') that are crucial to

the operation of the RESERV system to make sure that they are there. If they are not there, it chains to the program 'NEWYR' to create them and fill them with null characters. (The RESERV system uses virtual arrays open in UPDATE mode, so that all arrays must be pre-extended. This is a bother but allows several users to be putting books on reserve simultaneously at the beginning of the semester.) RESERV then indicates command readiness by printing the symbol '#', and waits for a user command. When a command is given, the first things looked for are the 'HELP' commands, which are basically any commands that start with the word 'HELP' or a question mark ('?'). It then checks to see if the command begins with 'E' ('END') or 'LOG' ('LOGOUT'). If the command fits none of these categories, RESERV checks the PPN of the job running against the PPN given by A2\$, and if these do not match it rejects the command immediately. If they match, RESERV continues scanning for valid command forms. First it looks for a number-prominent command. This can be a number alone (a reserve code) or it can be a number preceded by an 'R', 'N', or 'W' (for Return, Renewal, and Waiting list) and/or followed by a comma (or a slash) and then a string (the name or ID# of the person checking out the reserve or going on a waiting list. If the command fits this form in any way, RESERV enters a module to decide where to send the command. The preceding letter, at this point, is stored in G\$, and is "" if there was no letter before the number. The trailing string is V3\$ ("" if none), and the number itself is M2. RESERV first checks to see that M2 is a valid reserve code, then checks to see if it is currently checked out-- if it isn't checked out, return and renewal make no sense-- and then chains to 'INDESK', 'OUDESK', or 'DETAIL' depending upon the command (for checkout, checkin/renewal, and waiting list, respectively). If the command is not sufficiently specific, it will ask the user to choose return, renewal, or waiting list before chaining.

If the command is not in number-prominent form, RESERV will compare it with a list of stored keywords. If it matches, RESERV will put the command in core common and chain to the appropriate program, although in many cases it will require a password first. The structure here is fairly clear from the source files.

Chaining Structure

(All chains go both directions unless specified otherwise below.)

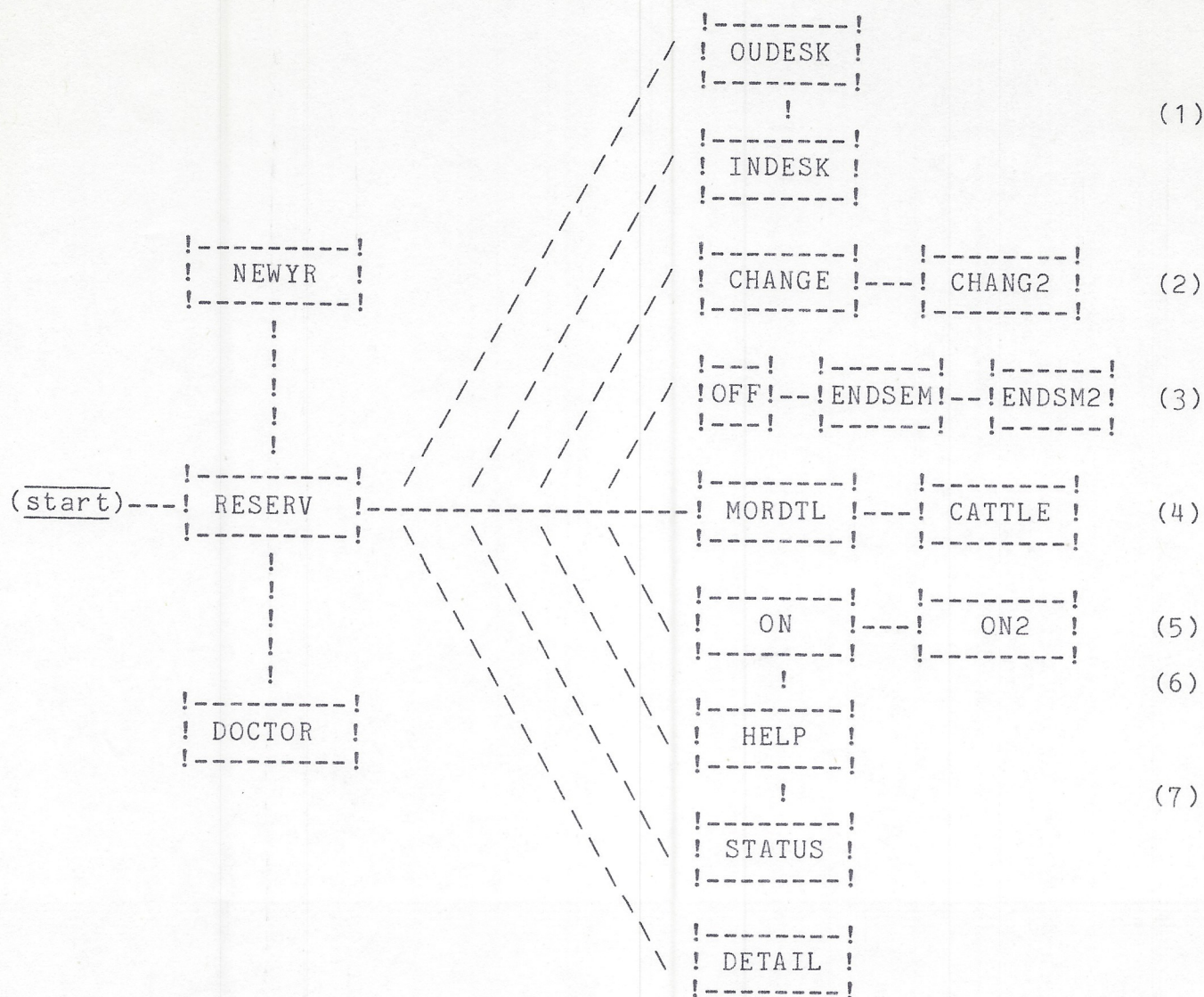


Fig. 1 -- Chaining structure of the RESERV system. Note that there are only 7 cases in which a program chains to any other program than RESERV:

1) OUDESK chains to INDESK when a book is being renewed (OUDESK returns it, INDESK checks it out again).

2) CHANGE chains to CHANG2 when the item to be changed is not a reserve reading but is actually a professor/course entry. (The two types of changing will not fit in a single 8K program.)

- 3) OFF chains to ENDSEM when, at the end of the semester, many reserves are being taken off reserve at once, including all reserves for specified professors. ENDSEM chains to ENDSM2 to finish the job.
- 4) MORDTL chains to CATTLE at line 25000 to request that a batch job be set up to run the CATTLE program from the beginning to create a certain kind of catalog. (MORDTL will also set up a batch job to run the CARDS program in response to the 'CARDPRINT' command.)
- 5) ON chains to ON2 and back to actually place a book on reserve after some preliminary information has been obtained. (The entire ON program will not fit in 8K.)
- 6) ON chains to HELP and back to get a listing of professors and courses that now have items on reserve (This is only done when the user types '?' in response to 'NAME OR NUMBER OF PROFESSOR/COURSE?')
- 7) HELP chains to STATUS to execute some of the HELP functions because there are eight such functions in all, and they could not possibly all fit in 8K.

Batch Structure

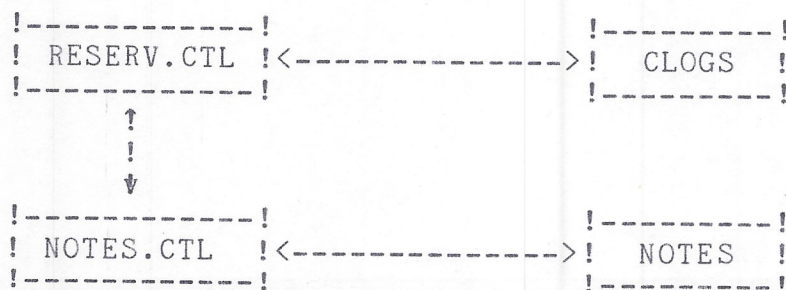


Fig. 2 -- Batch structure of the RESERV system. Note that two batch jobs run daily; when RESERV.CTL is finished, it queues NOTES.CTL to run the next morning. When NOTES.CTL is finished, it queues RESERV.CTL to run that evening. RESERV.CTL prints out catalogs, while NOTES.CTL prints out overdue notices. As noted previously, the MORDTL program will set up a batch job to run the CARDS program when the 'CARDPRINT' command is given, and will chain to the CATTLE program at line 25000 when certain cataloging options are selected. The CATTLE program will then set up the appropriate batch control file to run itself detached from the terminal.

VIII. Data Structure

The RESERV system was designed to store all string values dynamically in virtual arrays. Of course, virtual arrays are by nature fixed (and hence non-dynamic) storage facilities, so a referencing structure had to be established. It works like this: The reserve code is broken internally into two numbers, generally referred to as M% and N% (there are only a few exceptions). M% represents the number of thousands in the reserve code, and N% the remainder. Hence a reserve code of 32047 would break down as M%=32 and N%=47. These numbers flag the central record for the reserve. If M% is zero, then the reserve is a permanent reserve, and the information will be found in P\$(N%) on the file 'PERM.DAT'. Otherwise the information will be found in R\$(M%,N%) on the file 'LIB.DAT'. Since the data retrieval functions themselves always take care of deciding whether to look in 'PERM.DAT' or 'LIB.DAT' for the information, the rest of this explanation will deal only with the non-permanent reserves stored on 'LIB.DAT'. The case for permanent reserves is identical, only the names have been changed to protect the permanence.

The record R\$(M%,N%) is a sixteen character string, which is broken down into 8 integers in CVT%\$ format. All of these are accessed via the data retrieval and storage functions to be described shortly, so they need never be dealt with in the body of the program. One of these integers is a reference integer, that tells where to find the dynamically stored strings containing information on title, author, coding, and the name(s) of the professors who have put the item on reserve. That integer (let us call it I%) refers to a position in the T\$ array on the file 'LIB.DAT' (or the TO\$ array on 'PERM.DAT'). The strings for that reserve begin with T\$(I%) and continue with T\$(I%+1%), etc. T\$(0) contains only a number, the counter of how many places in the T\$ array have been used. The strings themselves begin with a null character. Thus when an entry in the T\$ array is reached that begins with a null character, it clearly refers to a new entry, and the previous string is complete. This entire string (of possibly enormous length, although not usually) is obtained and broken down into author, title, etc. by the data retrieval functions. Again, this breaking down need not usually be dealt with in the body of the program, but for the record, the structure of the string storage is as follows:

```
CHR$(0)+CHR$(1)+CVT%$(M%)+CVT%$(N%)+CVT%$(R%)
+CVT%$(A1%)+...+CVT%$(AR%)
+<AUTHOR$>+CHR$(12)+<CODING$>+CHR$(27)+<TITLE$>
```


where $M\%*1000+N\%$ is the reserve code, as described above, and $R\%$ is simply a counter that tells how many integers follow before the $\langle \text{AUTHOR}\$ \rangle$. Each of the integers that follows is the identifying number of a professor that has the item on his reserve list. (The professor/course name can be found in $J1\$(A1\%)$, etc.) This allows a reserve to be cross-listed for as many professors as is desired. Of course, $\langle \text{AUTHOR}\$ \rangle$ represents the author's name, etc....

The explanation of storage in other arrays is as follows:

$O\$(150)=64$, #5 ('CIRC.DAT') -- This is the circulation array. If reserve code $M\%*1000+N\%$ is checked out, then $FNR\%(M\%,N\%,\text{'STATUS'})$ is non-zero, and refers to the position in the $O\%$ array in which can be found information about the person who has it, when it is due, etc. Storage is of the form:

nn-xxx-mmddd aaabbbcccc\$\$\$\$\$\$\$\$.....

where nn-xxx-mm is the date due, dddd is the time due in military time, aaa is $M\%$, bbb is $N\%$, cccc is the ID# (possibly zero, if no ID# was given), and $$$$$.....$ is the name of the person who has the book. Storage is different in $O\$(0)$, where the first four digits are the number of $O\%$ entries that are used currently, and the remaining portion is the date that RESERV was last run. This date is checked each time RESERV is run, and if it is different from the current date then the RESERV program erases all waiting lists and prints out circulation statistics for the previous day, and then changes the right half of $O\$(0)$ to the current date.

$A5\$(15)=32$, #5 ('CIRC.DAT') -- This is simply the circulation array for keys. $A5\$(I\%)$ is blank if key # $I\%$ is in the library, and contains the name of the person who has it if it is checked out. $A5\$(0)$ contains the authorized library password.

$H\$(5)$, #2 ('LIB.DAT') -- Special counter array: $H\$(0)$ is the number of entries in the $J1\%$ array (see below), $H\$(1)$ is the number of pamphlets on reserve, $H\$(2)$ is the number of tapes on reserve, $H\$(3)$ is not used, $H\$(4)$ is not used, and $H\$(5)$ is the number of books checked out so far today.

$J1\$(500)=64$, #2 ('LIB.DAT') -- Each entry contains the name of a professor/course that has items on reserve. The number of items in this array is recorded in $H\$(0)$. $J1\$(0)$ is "PERMANENT RESERVE"

J1%(500), #2 ('LIB.DAT') -- Alphabetically references the J1\$ array. J1%(0) contains the number of entries in the J1\$ array. J1\$(J1%(i)) will contain the ith professor/course entry alphabetically.

S\$(1400), #4 ('STUDES.DAT') -- Entry #I contains the name of the student with ID#I.

S9%(1400), #4 ('STUDES.DAT') -- Alphabetically references the S\$ array. S9%(0) contains the number of entries in both the S9% and S\$ arrays. S\$(S9%(i)) will contain the ith student name alphabetically.

M%(1000), N%(1000), #10 ('CARDS.DAT') -- Contains the reserve code of books for which cards need to be printed out. M%(0) contains the number of entries that follow. Cards are printed for each reserve code=M%(I)*1000+N%(I) for I=1 to M%(0). Card requests are entered in the M% and N% arrays automatically by the ON and CHANGE programs, and also on request by the CAR module of the MORDTL program.

C%(1000,1), #9 ('PRFnnn.CAT') -- A .CAT file exists for each professor with items on reserve. If J1\$(17)="JONES", then the items on reserve for "JONES" are documented in the file 'PRF17.CAT', opened as file #9. ABS(C%(I,0))*1000+C%(I,1) is the reserve code of the I'th entry in the catalog. The absolute value is necessary because it may be negative. If it is negative, the entry is alphabetized by author, otherwise it is alphabetized by title. The C% array is maintained in alphabetical order by the CLOGS program. C%(0,0) is the number of entries that follow, and C%(0,1) is a special cataloging flag. If it is negative, that means that entries to the catalog have been added, changed, or deleted, so that the CLOGS program should re-alphabetize it. Then it will take the absolute value of C%(0,1) and print out that many copies of the catalog. Finally it will reset C%(0,1)=0. If C%(0,1)=0 when the CLOGS program runs, then the catalog does not need either re-alphabetization or reprinting, and the CLOGS program will do nothing.

Incidentally, the motivation for creating a separate catalog file for each professor is as follows: The RESERV system was designed to operate in UPDATE mode, allowing many users to be accessing data and putting things on reserve simultaneously. Unfortunately, this can lead to trouble when two people are putting things on reserve for the same professor. By putting cataloging information on separate files for separate professors, and by opening these files in normal (non-UPDATE) mode, it is possible to tell if anyone

else is putting items on reserve for the same professor, and it is easy to prohibit two people from putting items on reserve for the same professor.

Other Data Storage

FINES.DAT -- This file is a formatted ASCII record of outstanding fines. A new entry begins with the '@' character immediately followed by the name of the student who owes the fines. The following line contains only the amount owed. The lines that follow are simply comment lines explaining how the fine was accrued, and continue until the next line starting with '@' is encountered. This file is modified by OUDESK when a book is returned or renewed late, and by the '\$' or 'FINE' commands in response to '#'.

MESS.TXT -- This file is a formatted ASCII text message that is printed out with each overdue or fine notice. It is modified by the 'MES' command.

REALAB.DAT -- This file is simply a formatted ASCII list of all people authorized to use the reading lab, in no particular order. It is modified by the REA module of the DETAIL program.

***.WTL** -- These temporary files are created when someone wants to go on a waiting list. For example, if someone wants to go on a waiting list for reserve code #34007, then a file will be created by the name of '34007.WTL'. The file is formatted ASCII. The first line contains simply a number, which is -1 if the book is still checked out. If the book is not checked out, then the number indicates the time returned in minutes after midnight. (If it is just after midnight, it refers to minutes since midnight the previous day. Thus the number ranges from 120 to 1560.) Each remaining line of the file contains the ID# and name of a person on the waiting list. They are in order, and the ID# is followed by a comma. All files with a .WTL extension are deleted by the RESERV program the first time it is run daily. The standard test to see if there is a waiting list is simply to open the appropriate '.WTL' file for input. If no error occurs, then there is a waiting list, although the number on the first line must be checked to see whether or not it has expired.

Data Storage and Retrieval Functions

Actual array storage structure of information about reserve readings was described in the previous section, but should be largely unnecessary for modifying any part of the RESERV system other than the storage and retrieval

functions. These functions (FNR%, FNR\$, and FNI\$) are used almost exclusively to access information about reserve readings. In this section, we assume that the reserve in question has reserve code $M\%*1000+N\%$. Note that if $M\%=0\%$ then the reserve is permanent and hence the data is in a different file, but that does not affect the functions at all-- they know where to look no matter what.

The FNR% function has six options. FNR%(M%,N%,'TYPE') returns an integer from 1 to 8, coding the type of reserve (book, record, etc.). FNR%(M%,N%,'COPIES') returns the number of copies of this item on reserve. FNR%(M%,N%,'COPY') returns the copy # of this item (as in 'copy 3 of 7'). FNR%(M%,N%,'RESERVETIME') returns a number from 0 to 2. 0 indicates 2 hour reserve, 1 indicates 1 day reserve, and 2 indicates 3 day reserve. FNR%(M%,N%,'STATUS') is zero if the reserve is in the library; otherwise it indicates the position of the circulation information in the O\$ array. FNR%(M%,N%,'FREQ') indicates the number of times the reserve has been checked out this semester and is incremented each time the reserve is checked out.

The FNR\$ function is of the same form with five options. FNR\$(M%,N%,'TITLE') returns the title of the reserve. FNR\$(M%,N%,'CODE') returns the coding of the reserve. FNR\$(M%,N%,'AUTHOR') is the author, while FNR\$(M%,N%,'PROFS') returns a string containing the names of all professors for whom the item is on reserve, separated by commas. FNR\$(M%,N%,'STRINGS') simply returns the unaltered string containing all of these records as stored in the T\$ array.

Finally, the FNI\$ function is used to place new values on file for a reserve. The general form is FNI\$(M%,N%,Q\$,Z\$) where Z\$ is the new value and Q\$ is the label. Valid labels include all of the FNR% labels ('TYPE', 'STATUS', etc.) as well as the label 'STRINGS'. Note that FNI\$ can not be used to insert titles, etc., alone. The entire string must be prepared as it is to be stored in the T\$ array. For a simple example of this, see Option #3 of the DOCTOR program. Also, please note that the new value is defined to be a string (Z\$) in all cases. Thus if the new value for the STATUS of reserve code $M\%*1000+N\%$ is 0% (i.e. the book is being checked back in), then the correct statement is $X\$=FNI\$(M\%,N\%,'STATUS',CVT\%$(0\%))$. Numbers must be passed to the FNI\$ function in CVT%\$ format!

There is only one special flag making use of the function calls. FNR%(M%,0%,'TYPE') returns the number of reserves whose first number is M%. This is used in assigning new reserve codes and in calculating occupied space (the last part of the STATUS program and DOCTOR option # 3.)

IX. Programming/Maintenance Aids

The primary programming aid written into the RESERV system is the DOCTOR program. The program does nothing at all for the RESERV system itself, but it contains every option of every function call, which is a handy reference, and it contains three options when accessed via the normal 'RUN' command. If you choose the first option, the program will exit to immediate mode without printing the READY message and without closing any files. This allows immediate mode operations on all virtual arrays. Typing 'CONTINUE' at this point will CLOSE all files and chain back to the RESERV program.

The second option allows testing of the string storage and retrieval function. You give it the author, title, coding, and all of the professors numbers for whom you want this information stored, and it will store the information and then retrieve it again. This allows you to make sure the functions are working properly, and perhaps more importantly, the ten lines of source code for this module provide the best possible illustration of how these functions work.

The third option simply reports on remaining space in virtual arrays. It tells you what percentage of the pre-extended space is being used. If any of these percentages rise above 100%, the RESERV system will crash.

Another programming aid worth mentioning involves the way in which the ON2 program searches a reserve's title for leading articles ('a', 'an', 'the', etc.) so that the reserve may be alphabetized properly. All of the articles it searches for are set in a DATA statement, so that if the library ever thinks of more articles that should be truncated but are not currently being truncated, they can simply be added to the data given in the DATA statement.

The last programming aid worth mentioning involves the formatting of card printout. The CARDS program was designed to be easily modified to print library cards on different types of forms. All you have to do is change the four size parameters that are set at the beginning of the program regarding form length, form width, etc. This is explained in comments at the beginning of the CARDS program.

Source programs for all RESERV system programs are stored in the file RESERV.LIB in the storage (A1\$) account. They may be accessed via &LIB. Good luck!!

X. Appendix: Summary of RESERV-MAC commands

The following are simple responses to the number sign (#):

'K7'-- Checks key #7 in or out, depending on its current status
'1001'-- Checks out reserve #1001 if it is currently in the library and allows you the options of return, renewal, or waiting list, otherwise
'1001,JOHN DOE'-- Checks reserve #1001 out to John Doe
'R1001'-- Returns reserve #1001
'N1001'-- Renews reserve #1001
'W1001,JOHN DOE'-- Tells the computer that John Doe wants to go on a waiting list for reserve #1001
'FIN' or 'FINE' or '\$'-- Tells the computer than someone wants to pay a fine
'E'-- Ends the use of the reserv system
'LOG'-- Logs you off the computer

The following commands may be used to request information:

'?1001'-- Tells you all about reserve #1001
'?P1'-- Lists all reserves for professor #1
'?P?' OR '?PROF'-- Lists all professors who now have reserves
'?JONES,HENRY'-- Lists all reserves for professor Jones, Henry
'?MATHEMATICS'-- Lists all reserves for the Mathematics department
'?ATH/THEORY'-- Lists all reserves for the Math dept, Prof. Rather, etc. that contain 'THEORY' somewhere in the author or title
'??'-- Prints out a copy of the complete documentation on these programs
'STA' or 'STATUS'-- Lists all reserves and keys currently checked out
'?' or 'HELP'-- Gives you the choice of any of the above

The following special commands require a password:

'ON'-- Puts books on reserve
'OFF'-- Takes books off reserve
'CHA' or 'CHANGE'-- Changes erroneous information about reserves
'CAT' or 'CATALOG'-- Requests that new catalogs be printed out
'CAR' or 'CARDS'-- Requests that new cards be printed up when the old one is lost or otherwise disenabled
'MES' or 'MESSAGE'-- Changes the message printed on overdue notices
'PAS' or 'PASSWORD'-- Changes the password used to run these programs
'ADD'-- adds a new student to the roll of students
'REA' or 'READINGLAB'-- Authorizes a student to use the reading lab
'AUTO'-- Restarts batch jobs for catalogs and overdue notices when these have been stopped
'STOP'-- Stops batch job automatic preparation of catalogs and overdue notices

XI. Index

\$CHANGE	19
\$QUE	19
\$SORT	19
ADD	12
AUTHOR	28
AUTO	17
Adding new reserves	12
Alphabetization of titles . .	14
Array storage	25
Articles	14,29
Author Catalogs	11
Author/title catalogs	11
Automatic printing control . .	17
Backup cards	17
Batch jobs	17
Batch structure	23
Benefits	3
CARDS	11
CATALOG	10
CHANGE	10
CLOGS	26
CODE	28
COPIES	28
COPY	28
CTRL/C	3
CTRL/U	3
Card backups	17
Card parameters	29
Card size modification	29
Cards	11,19
Cards array	26
Catalog arrays	26
Catalogs	10
Chaining structure	22
Changing Professor/course entries .	11
Changing card parameters	29
Changing message	11
Changing password	11
Checkout	8
Circulation array	25
Command parsing	20
Command summary	2
Command syntax	20
Commmand summary	30
Computer errors	4
Contents	2
Correcting erroneous data	10

RESERV-MAC: Burling Library Reserve Book System

Correcting typing errors	13
Counters	25
Crosslisting	13
Cursor controls	19
DELETE	3
DOCTOR	29
Data retrieval	27
Data storage	27
Data structure	24
END	7
ENDSEM	23
ERRLOG.DAT	4
ESCAPE key	13, 16
End of semester routines	15
Error procedures	4
FNI\$	27
FNR\$	27
FNR%	27
FREQ	28
Fine list	9
Fines	9, 9, 27, 27
Finishing	7
Function testing	29
Getting Started	5
HELLO	5
HELP	6, 23
INDESK	22
Immediate mode operations	29
Internal counters	25
Introduction	3
Keyboard	3
Keys	9, 25
LIB.DAT	24
LOGOUT	7
Library card size	29
List of courses	7
List of fines	9
List of professors	7
List of reserves	6
Lists	6
Logging out	7
MESSAGE	11
Messages	5, 27
Missing files	21

NEWYR program	21
Number-prominent commands	21
OFF	10,14,23
OLD	13
ON	10,12,23
ON2	23
OUDESK	22
Occupied Space	29
Outstanding fines	27
Overdue books	9
Overdue message	27
Overdue text message	11
Override	16
PASSWORD	11
PERM.DAT	24
PIP	17
PROFS	28
PROGRAM TERMINATED message	4
Parsing	20
Password	5,25
Paying fines	9
Periodicals, volume number	14
Professor/course entries, changing	11
Professor/course list	7
Professorial array	25
Publicly accessible commands	6
Putting on reserve	12
QUE	17,19
REA command	11
READY	5
RESERV	3
RESERVETIME	28
Reading lab	11,27
Removing reserve items	14
Renewal	8
Requesting cards	11
Requesting catalogs	10
Reserve code	24
Reserve lists	6
Restarting batch jobs	17
Retrieval functions	27
Return	8
SORT\$	19
STATUS	6,23,28
STOP	17
STRINGS	28
SYS calls	19

Security	19
Semester end routines	15
Separation of articles from titles 14,29	
Shelf lists	11
Special features	17
Special forms	19
Starting	5
Statistics	11,17
Status report	6
Stopping batch jobs	17
Storage functions	27
Storage of strings	24
String storage	24
Student array	26
Summary of Commands	2,30
System generation	21
System shutdown	4
TITLE	28
TYPE	28
Taking off reserves	14
Technical overview	19
Text message	27
Title Catalogs	11
Title alphabetization and article separation 14,29	
Typing errors	13
Unanticipated Error(s)	4
Volume number of periodicals 14	
Waiting list	8
Waiting lists	27
XLIST	13
Yes-or-no	3

RESERV-MAC:

Reserve expandable system enhancing
Record verification, management, and circulation

The RESERV-MAC system is a system of twenty computer programs written in the BASIC-PLUS programming language for implementation on the Digital Equipment Corporation PDP 11/70 computer. The programs are designed to occupy a maximum of 8K core storage area to run, although the modularity of design and the specific functions of each program are made invisible to the user via 'CHAIN' statements by which one program passes control to another.

The purpose of the RESERV-MAC system is to streamline the operation of the reserve desk by automating some of its key functions. RESERV-MAC is capable of producing several different types of catalogs, as well as library cards and overdue notices. In addition it provides limited on-line bibliographic search facilities for anyone with an account on the college computer, and provides the library staff with continually updated information regarding reserve book usage. RESERV-MAC has made feasible the production of overall shelf lists and integrated alphabetized catalogs to supplement the traditional course-oriented reserve lists, and has permitted the regular issuance of hitherto prohibitively costly statistical summaries of reserve usage.

The operation of the RESERV-MAC system is relatively simple, requiring a bare minimum of training. To check a book out under the current version of RESERV-MAC, a library worker types the reserve's identifying number, then a comma, and then the name or ID# of the student checking it out. (Obviously this procedure could be greatly streamlined by the purchase of new hardware and the production of machine-readable ID cards and library cards, but no such expenditure has yet been made.) The computer will make sure that the book is available (i.e. the book is in the library and either there is no waiting list or the waiting list has expired) and will then announce the date and time due. To return a book, the librarian merely types the letter 'R' followed by the book's identifying number. If the book is overdue, the computer will ask if the fine has been paid, and will deal with the matter accordingly. (If the fine is unpaid, the student will receive fine notices twice weekly until it is paid.)

Other reserve functions vary in operating complexity, although training someone to use the entire system in all its aspects takes only a few hours. Available commands, in addition to those for circulation of reserves and library keys, include 'STATUS', for a status report on all items checked out, 'HELP', for a smorgasbord of assisting functions, and several simply formatted information queries. For example, typing '?MATH' obtains a list of all reserves for the Mathematics department, while '?SMITH' will get a list of all reserves for any professor named Smith. '?MATH/STAT' provides a selective search, listing all

reserves for the Math department that contain 'STAT' somewhere in the author or title. '?1001' will get a detailed report on everything there is to know about reserve number 1001, while '?PROFS' will get a list of all professors with items on reserve. The user can get lists of available commands by typing '?' or 'HELP' and choosing one of the lists from among the eight types of help he is offered. All of these information queries are available to any user of the Grinnell College timesharing system.

Many other commands are, of course, available only to the librarians. One of these is the 'ON' command, for putting items on reserve. In this program, the user enters the information regarding the reserve at a cathode ray terminal. The data entry takes the form of a dialog; the computer asks for the title and the user enters it, and so on, so that there is no need for the user to remember any special format for data entry. Also, the ESCAPE key can be used to go back to the previous question in the event that an error has been made in the entry of the data. When all data has been entered, the computer assigns the reserve number, and marks it internally to have the library card and appropriate catalog pages printed out.

The problem of printout frequency, which has plagued so many library systems with the alternatives of a deluge of paper or a substantial lag between data entry and catalog production, is also a problem in the RESERV-MAC system. The current idea is to update each professor's list as it becomes necessary; thus if a new item is put on reserve for Professor Jones, then a new copy of his reserve list will arrive in the campus mail the next morning. The librarians also have at their disposal a 'CATALOG' command. When given this command, the computer responds by giving the librarian the choice of several options, to produce comprehensive shelf-lists, author indices, title indices, combined author/title catalogs, or extra copies of the professorial catalog for all or some professors. Of course the user is allowed to specify how many copies he wants of each. In this system the professors' lists are updated daily, but the larger catalogs are updated only when the reserve workers think the update is necessary. The latter scheme is also used for the production of library cards; every morning, when the librarian logs into the reserve system, the system will, after announcing how many items were checked out the previous day, et cetera, will announce how many cards are currently awaiting printing. The librarian can at any time request that all cards awaiting printout be actually printed by giving the 'CARDPRINT' command. The librarian can also request that items be added to the list of those requiring card printout by the 'CARD' command.

RESERV-MAC will accept many other commands, too many to mention in this article. The user can specify a text message to accompany overdue notices, override waiting lists, pay or override fines, take items off reserve, alter information about items on reserve, change passwords, modify the list of students authorized to use certain keys, stop or restart the automatic production of catalogs and overdue notices, and so on.

Evaluation

RESERV-MAC has proven useful as a tool to increase efficiency and service in the processing of reserve materials. Although it has added a few seconds to the length of time necessary to check items in or out, it has greatly reduced the amount of labor necessary for reserve paperwork. Such savings, if extended to the entire collection, would be very substantial, especially if an interface could be developed with the OCLC system now used by the college to catalog new acquisitions. RESERV-MAC in its current implementation can accomodate up to 10,000 reserve items. A modified version of the system, now in the planning stage, will expand this capacity to nearly 33,000 and will itself be expandable by very simple modifications to handle a collection of indeterminate length. Many RESERV-MAC features are strongly oriented towards specific features of Burling Library at Grinnell College, for example the circulation of the keys to the various rooms in Burling. However, the future designers of a general-case library automation system might wish to consider the various functions of RESERV-MAC and their generalization to other libraries. For Grinnell's library, most features of RESERV-MAC have proven handy and time-saving, and would be missed if omitted in future library systems. Copies of the complete RESERV-MAC Documentation manual are available from Grinnell College Library, Grinnell, Iowa 50112.