

B.A. Harper

SPECS: A MENU CONTROL SYSTEM FOR RT-11

B. A. Harper
Public Works Department N.S.W.
Hydraulics Laboratory
Manly Vale, Sydney

ABSTRACT

The basis of a menu structured software control system for an RT-11 environment is presented. The system is based on a FORTRAN program which interprets commands given in user created ASCII menu files, presents various options to the user and acts on the selections made. The system has a rich command set and its concepts are equally well suited to commercial uses and extension to other operating systems such as RSX and VMS.

INTRODUCTION

SPECS (you need a pair of specs to read a menu...) is a controlling/scheduling program (1) with functions which are user-selectable from within "soft" menu control files. Written in humble FORTRAN, it provides the basis for a menu structured software environment under RT-11.

All software designers share the common problem of how to establish a suitable user interface to complex software systems, especially where prospective users are relatively inexperienced. The "menu" concept offering a number of discrete options is often a good solution because of the opportunity perhaps to provide descriptions of each action and rigorously control the sequencing of activities. In addition, program communication and interlinking can also be effectively handled using a menu approach and result in considerable savings in software design. The inherent modularity of a menu structure means that changes in software components can be easily accommodated without affecting other sections.

SPECS has grown out of a need to make program systems appear more integrated than they really are, especially to inexperienced users, but also to reduce the costs of a fully integrated software system. Since its initial implementation some eighteen months ago, a number of additional features have been added so that instead of "coming to the rescue" of a complex software system it is now seen as forming the basis of such systems at the start of the design phase. Figure 1 illustrates schematically the broad unifying capability of SPECS.

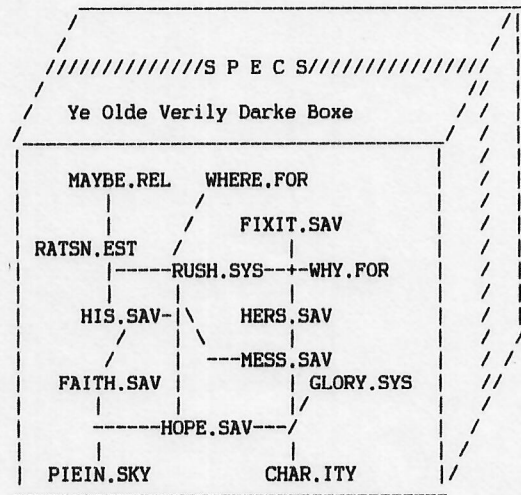
DESIGN FEATURES

General Operation

SPECS operates as either a stand-alone background or foreground program under RT-11 or as a subroutine with optional user-supplied command processing. Overlaying is used where possible to minimize memory usage.

An integral part of SPECS's operations involves the use of a LINK storage area 60 words in length which, during an RT-11 CHAIN operation, can be used for the exchange of information between background programs.

SPECS reserves use of the first 39 words of the LINK with words 40 to 60 being available for general use.



*Box courtesy Pandora Pty Ltd

Figure 1 SPECS To The Rescue

Menu Files

SPECS relies for its operation on at least one user supplied ASCII "menu" file being present (known as the MASTER menu) and as many other menu files as necessary to describe the system. The MASTER menu has the default filename DK:MASTER.MNU but other names may be specified at run time. Each of these files contains an individual menu which is interpreted by SPECS and then presented to the user to await a selection as shown in Figure 2. Parts of each file are reserved for control information describing to SPECS the types of actions to be taken for each option offered. This control information is not presented to the user but the remainder of the file, which includes a menu title and various options and their descriptions, are displayed "as is" which allows easy design of each menu layout.

When the user makes a selection the option is first checked for validity and then the specified action is taken, e.g. chaining to another program. Only one menu is in effect at any one time.

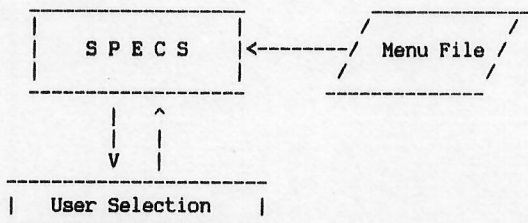


Figure 2 Basic Operation

Menu Tree Structure

Any number of sub-menu files may be logically linked to the MASTER menu in a tree-like structure. This is achieved by each menu nominating the menu filename which lies immediately above it in the logical structure (its TREE menu), but any degree of cross-linking is also allowed between menus regardless of their position within the tree. Figure 3 illustrates this system which in effect allows infinite nesting of menus. COMMANDS specified within each menu then allow MOVES from one menu to another, RETURNS to the last menu used, BRANCHING to a new menu after an action is complete, moving back up to the TREE menu or returning to the MASTER menu.

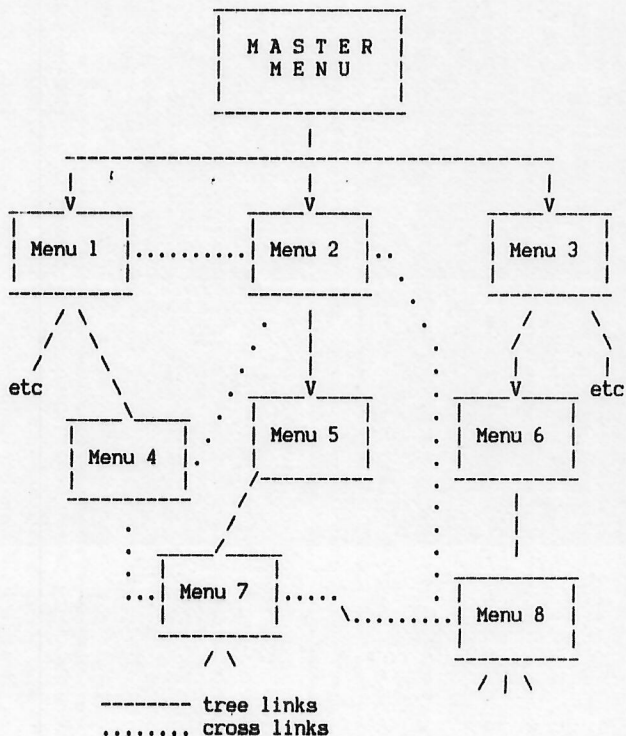


Figure 3 General Tree-like Structure

Menu File Format

Figure 4 shows the contents of a simple menu file whose structure is discussed below and Figure 5 shows

the user view as it is presented to the terminal. Fixed format inputs are used extensively to reduce parsing overheads and with a screen editor such as KED this is no handicap to efficient set-up.

```

User identification line/ date/ filename etc
[VDY1:TREE.MNU ]TITLE OF THE MENU
[HDYO:MENU3.HLP ] Menu Subtitle Description

[BDYO:MENU4.MNU ]1 Option 1 TEXT Description
[CFRED.SAV ]
[SDYO:DIRECT.COM]2 Option 2 TEXT Description
                2nd line of TEXT
                3rd line of TEXT

[DDY1:LINK.DMP ]3 Option 3 TEXT Description
[NDY1:OPT103.HLP]
[HDYO:HELP3.HLP ]
[CDK:EXEC3.SAV ]
  
```

Figure 4 Sample Menu File

- (i) The first line is reserved for user identification only and is not read by SPECS.
- (ii) The second line is always reserved for
 - (a) Video/Non-video toggle for display control
 - (b) The logical TREE menu filename
 - (c) The menu TITLE line (up to 64 characters).
- (iii) All subsequent lines in the file are interpreted as follows:
 - (a) The first 17 characters of each line are reserved for the SPECS COMMAND field to be acted upon in the event of a particular option being selected.
 - (b) The 18th character, if non-blank, is any ASCII character to be offered as an OPTION in the menu.
 - (c) Subsequent characters beyond the 18th and up to the 80th are always deemed to be a TEXT description of the option specified. This can extend to any number of lines as long as there is a blank in column 18.
- (iv) More than one COMMAND can be specified for each option, commands being processed in logical order "down" the menu until another option is specified (i.e. non-blank column 18) or EOF is reached. The series of commands pertaining to any option is referred to as its command "stack".
- (v) COMMANDS may also be inserted immediately below the TITLE line to be executed at the time the menu is displayed.
- (vi) General formatting of the TITLE and TEXT beyond column 18 in each line is at the user's discretion.
- (vii) In the special case of the MASTER menu the area normally reserved for the TREE menu filename is used for setting many of the special SPECS mode controls to be discussed later.

COMMAND Format

All SPECS commands are single ASCII characters (defined in a later section) followed by a 14 character argument field, the entire command field being contained within formatting delimiters "[" and "]" for readability. The width of the argument field

is expressly designed to accept a full RT-11 file specification of the form "dev:filnam.typ", since this is the most common command argument used in practice. The COMMAND field must always begin at the start of a line.

OPTION and COMMAND Stack

The number of OPTIONS and associated COMMANDs in any menu file is only limited by the size of internal buffers and is normally set to practical requirements such as the number of lines on a video display. In the menu of Figure 4, the lengths of command stacks are as follows:

menu stack	1	option 2	1
option 1	2	option 3	4

TITLE OF THE MENU

```

Menu Subtitle Description

1 Option 1 TEXT Description
2 Option 2 TEXT Description
  2nd line of TEXT
  3rd line of TEXT

3 Option 3 TEXT Description

?
```

Figure 5 User's View of Menu

LINK Data and Re-entry Capabilities

One of the basic commands available in SPECS is the ability to CHAIN to any nominated background program. As previously mentioned, LINK data is transferred to the chained program during this process which contains:

- the name of the SPECS program to allow a back-chain to resume menu selection
- the name of the menu file just used
- the option selected in that menu file
- the name of the menu to return to when back-chaining to SPECS
- assorted status control information
- up to 20 words of user-specific data

This allows the chained program to respond differently to various menus and options or to test user specified control information in the LINK data area. In addition, the name of the menu file to be returned to can be altered in the LINK data before re-chaining to SPECS. Figure 6 illustrates a typical CHAIN process and the basic form of a chained program is given below:

```

PROGRAM BGRND
BYTE LINK(120)           !Allocate space
C
CALL RCHAIN(IC,LINK,60)  !Check if CHAIN
. . .
. . .                   !Local processing
. . .
IF(IC,LT,0)CALL CHAIN(LINK,LINK,60) !Back-CHAIN
C
STOP
END
```

An analogous feature to chaining is available through the ability to send RT-11 keyboard commands to KMON when SPECS is exited. Such commands must be in the form of an indirect command file where the final command in the file normally re-runs SPECS. This allows considerable freedom in the use of monitor

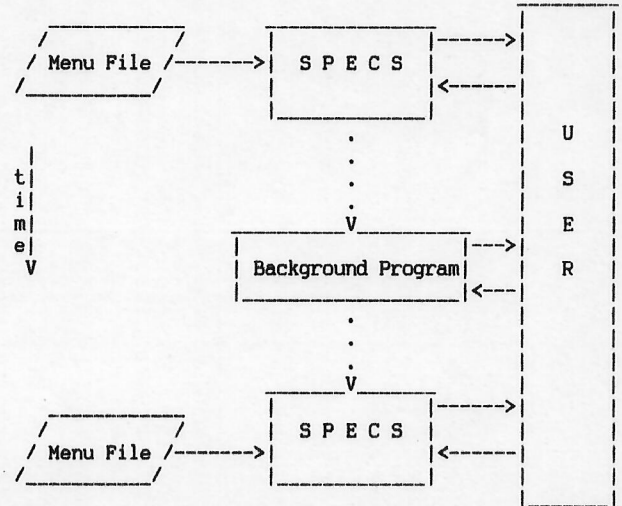


Figure 6 Typical CHAIN Procedure

level commands such as DIR, TYPE, PRINT etc., as well as RUN, FRUN and SRUN to schedule background, foreground or system jobs. When commands are to be sent to KMON the current LINK data is preserved on a temporary disk file. On later re-entry to SPECS it always checks to see if it was chained to, or that the temporary file exists. In the latter case the LINK data is then read from file and the temporary file deleted. This means, for example, the last menu file used can always be returned to, rather than the MASTER menu. When first running SPECS and a definite start at the MASTER is required, a utility program CLEAN is available to first purge the temporary file if it exists. Figure 7 illustrates a typical chain via KMON.

Other facilities allow LINK data to be explicitly DUMPed to file or INPUT from file as desired and to modify LINK data as a result of an option being selected. LINK data may also be displayed on the terminal or new values input from the terminal.

SPECS as a Subroutine

Two versions of SPECS are available for use as a subroutine. Firstly, the full implementation of the stand-alone system offers all the standard features plus the ability for the user to handle his own command processing (i.e. define his own command set). The second implementation, known as LENS (Local Environment Specs), is of a much reduced capability designed for uses where a "SPECS-like" user interface is required for local program processing, e.g. replacing a question and answer dialogue in a chained program. In this case a menu file is not used but rather the menu description is specified in a data array via the subroutine arguments and all processing is local only.

HELP Facilities

An inbuilt help facility is available to the user both at the menu level and at an individual option level. This is achieved through special ASCII help files whose names are specified in the command field of the menu adjacent to the respective option. HELP is then invoked via the user entering <escape><option>, rather than the normal selection of <option> alone. If a help file has been nominated for that particular option anywhere within its command stack then that file is presented to the user for his information. Return to

the menu file is then effected at the user's convenience. To invoke help for a particular menu <escape><escape> is used.

In addition to the HELP facility it is also possible to explicitly cause information to be displayed by using the NOTIFY menu command and the PAUSE command. The latter command, having displayed the nominated information to the user, waits for the user to give a YES/NO answer to either continue further command processing or else abort the option.

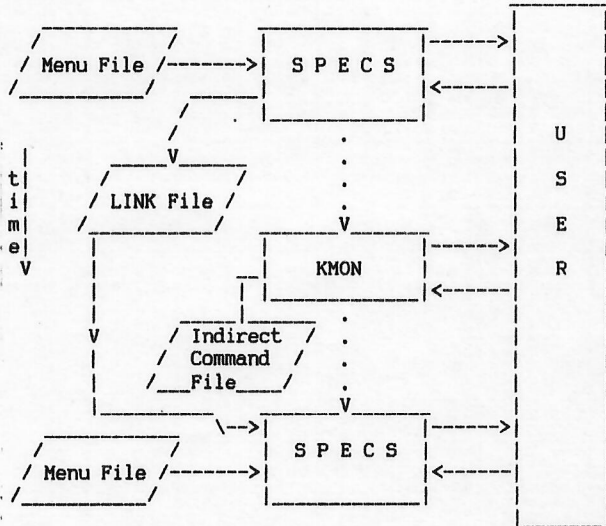


Figure 7 Typical Chain via KMON

VT100 Control

Several levels of VT100 control are available and these are selectable via status settings within the MASTER menu as follows:

- (i) Video or non-video menu declaration which simply invokes or cancels VT100 protocol on a menu by menu basis. Video mode must be set to enable features (ii) and (iii) below.
- (ii) Double height and width menu TITLE line.
- (iii) Video attribute flagging of the TITLE and TEXT as well as within associated display files such as used by HELP.

Video attribute flagging allows the specification of a special attribute flag character imbedded in the text which is followed by a 2 digit integer video display mode, e.g. if "%" is the attribute flag character then "%04make this line blink"

will cause a blinking display on output. If the advanced video option is fitted there is no limit to the number of display mode changes on any line to allow highlighting in reverse video, bold, underline or combinations of these displays (see DECUS 11-412).

If a particular menu file contains the attribute flagging but is not being displayed on a VT100 then the menu can simply be made non-video and all attribute flagging is automatically stripped prior to output.

Option Select Modes

SPECS can be configured to wait for <CR> terminators for option selection or to operate in immediate input mode (default).

Error Handling and Recovery

All illegal option selections are flagged by <bell> and erased (in video mode). If any error occurs during menu file access an automatic return to the last menu is initiated. In addition there are over 30 error messages, all of which "soft" fail to the present menu.

COMMAND Control Feature

A simplified "IF THEN SKIP" facility is available which, depending on the result of a logical test with a nominated LINK byte, will enable skipping over a number of following commands in the option stack. This allows different actions to be taken from the one menu option dependent upon a user designed status system.

Communication Features

From within a menu it is possible to establish communication between any job type e.g. background, foreground or system. Messages sent or received by SPECS may be synchronous or asynchronous and are stored within the user LINK area. The length of message is variable up to the full 20 words available.

COMMAND Subroutines

Where a long and complicated set of menu commands is needed or a number of commands are used by more than one option or menu, then the commands may be stored on a separate file and invoked at the appropriate stage in the option processing.

Earlybird Facility

If an option is chosen before the menu is fully displayed, further output is bypassed and the option is immediately executed. This is useful for long menus and slow storage devices.

Truncated Menu Capability

This feature optionally allows a shortened form of menu displaying two lines of output only. This consists of the menu TITLE and the list of available OPTIONS on the next line. This is particularly useful for hardcopy devices where paper usage is a problem and for restricted display devices such as liquid crystal. Truncate mode can be set either at the MASTER level, is selectable explicitly via a menu command and can be invoked manually at any time through a hidden option (see later section). All standard features such as HELP are still available but video mode is disabled. The menu of Figure 4 would appear as follows in truncate mode:

```
Menu : TITLE OF THE MENU
Select: 1, 2, 3 ?
```

Browse Facility

This feature is an aid to the truncated menu user who has forgotten what the various options actually mean! It only operates in truncate mode where, by entering a single <CR> instead of <option> or <escape> etc., the menu TEXT can be browsed through one line at a time. Each time <CR> is entered the next line of TEXT is displayed and <option> or <escape> can still be entered at any time. Browsing automatically wraps around from EOF to SOF when appropriate. Browse mode can also be invoked during HELP, NOTIFY and PAUSE but no file wrap-around applies.

Additional features allow either the menu title to be repeated at each <CR> as well as the next line of text or only the displaying of TEXT which has an OPTION.

Debug Mode

A special debug mode can be entered manually at any time and allows entry of any standard menu command such as might appear in a menu file. For example this would allow moving to another menu file even though the present menu does not have that option.

Hidden Options

SPECS has a number of reserved or "hidden" options as follows:

^D enter debug mode (if enabled in MASTER menu)
^T truncate menu display toggle
^W redisplay present menu
^ (caret) go back to TREE menu
* return to MASTER menu
<esc> invoke HELP facility, followed by <option>

Control-C Handling

A number of options are available for dealing with ^C inputs, as follows:

(i) Enable normally as per RT-11
(ii) Disable and ignore at all times
(iii) Emulate "*" option and return to MASTER
(iv) Emulate "^" option and go back to TREE
(v) Ignore when at MASTER

These options can be set in the MASTER menu.

COMMAND SET

Table I lists the complete COMMAND set in alphabetical order, which includes upper case A-Z and the special characters "^" and "*" which mimic their hidden option equivalents. Most commands expect a 14 character argument (generally a file specification but some require specialised arguments) while others do not require an argument at all.

TABLE I - SPECS Command Set

A	ACCEPT LINK data from terminal
B	BRANCH to new menu after processing
C	CHAIN to a background program
D	DUMP LINK data to a file
E	EXIT to KMON
F	not yet allocated
G	not yet allocated
H	HELP file specification
I	INPUT LINK data from file
J	JUMP commands in stack
K	not yet allocated
L	LINK byte modification
M	MOVE to menu file
N	NOTIFY the user
O	OUTPUT LINK data to terminal
P	PAUSE with message to user
Q	QUEUE message for another job
R	RETURN to last menu
S	SEND indirect command file to KMON
T	TRUNCATE menu display
U	USER action required
V	not yet allocated
W	not yet allocated
X	EXECUTE subroutine COMMAND file
Y	not yet allocated
Z	not yet allocated
^	Go back to TREE menu
*	Go back to MASTER menu

For clarity, the full command set will be discussed in terms of two subsets viz. the BASIC and ADVANCED commands. The BASIC commands all perform a logical "end of processing" function as part of their execution while the ADVANCED do not.

BASIC Commands and Their Format

CHAIN to a background program [Cdev:filnam.typ]

Terminates present command processing and begins execution of the program specified as a background job.

EXIT to KMON [E]

Terminates present command processing and unconditionally exits to KMON if standalone. If subroutine then returns to calling routine with a null option.

MOVE to menu file [Mdev:filnam.typ]

Terminate present command processing and transfer control to the menu file specified.

RETURN to last menu [R]

Terminate present command processing and effect a MOVE to the last used menu file.

SEND indirect command file to KMON [Sdev:filnam.typ]

Terminate present command processing and send filename to KMON for execution as an indirect command file after SPECS exits. LINK data will be DUMPed to a temporary file.

USER action required [Udev:filnam.typ]

Terminate present command processing. If in stand-alone mode execute a CHAIN command but if in subroutine mode check dev:filnam.typ against name of calling routine and if same then perform a subroutine return.

^ Go back to TREE menu [^]

Terminate present command processing and effect a MOVE to the name of the TREE menu specified in the present menu. If the TREE name is given as "*" then move to the MASTER menu. If already at the MASTER menu then, depending on the EXIT status setting, either exit to KMON or ignore command.

* Go back to MASTER menu [*]

Terminate present command processing and effect a MOVE to the MASTER menu. If already at the MASTER menu then, depending on the EXIT status setting, either exit to KMON or ignore command.

ADVANCED Commands and Their Format

ACCEPT LINK data from terminal [ALnnn,mm=(ffff)]

Prompt the user for input with "A", accept a line of input terminated by <CR> and decode it according to the FORTRAN FORMAT ffff. Store the mm bytes of data starting at LINK byte no. nnn, e.g. [AL095,10=(10A1)] .

BRANCH to new menu after processing [Bdev:filnam.typ]

After processing a C, S, or U command take the

filename specified as the name of the next menu file instead of automatically returning to the present menu file.

DUMP LINK data to a file [Ddev:filnam.typ]

Write out the 60 word LINK data area to the file specified in unformatted binary format.

HELP file specification [Hdev:filnam.typ]

Use the filename specified if the user invokes the HELP facility and display its contents starting at the 2nd line of the file. The command may be specified anywhere within a particular option's command stack or between the TITLE line and the first option when used for menu help.

INPUT LINK data from file [Idev:filnam.typ]

Reverse process to DUMP.

JUMP commands in stack [JccLnmntsssssss]

Jump over the next cc commands in the stack if the result of test t between LINK byte nnn and ssssss is .TRUE., where ssssss is an ASCII character of the form 'A', an integer between ±32767 or another LINK byte Lnnn and t is a test of the form =, /, >, <, e.g. [J02L105>L104] .

LINK byte modification [Lnn=ssssssss]

Transfer the data specified in ssssssss to the LINK data area starting at byte nnn, where ssssssss is an ASCII string of the form 'ABCDEFGH', an integer between ±32767 or other LINK data specified as Lnnn,mm, e.g. [L100='message'] .

NOTIFY the user [Ndev:filnam.typ]

Display the contents of the specified file on the terminal starting at the 2nd line and wait for the user to acknowledge before continuing processing.

OUTPUT LINK data to terminal [OLnnn,mm=(ffff)]

Prompt the user prior to output with "O>" and output mm bytes starting from LINK nnn in a FORTRAN FORMAT according to ffff, e.g. [O095,10=(10A1)] .

PAUSE with a message to user [Pdev:filnam.typ]

Perform a NOTIFY but force user to give a YES/NO reply to indicate processing should either continue or abort.

QUEUE message for another job [Qxhj=ssssssss]

Queue the message ssssssss which is an ASCII string of the form 'ABCDEFGH', an integer between ±32767 or LINK data of the form Lnnn,mm according to x either S (send) or R (receive) and h either W (wait) or N (no wait) with jobname j either B (background), F (foreground) or 6-1 (system), e.g. [QSW6=L100,07] .

TRUNCATE menu display [TON]
[TOFF]

Enable or disable truncated menu display.

EXECUTE subroutine COMMAND file [Xdev:filnam.typ]

Continue processing of the present command stack at the 2nd line of the file specified and at EOF return to processing the next command (if any) in the present menu command stack. Processing may be terminated within the subroutine file if desired.

APPLICATIONS

Three major applications of SPECS are briefly reviewed below to show the versatility of the command set.

Laboratory Experiment Control

SPECS was originally developed as an aid for experimental control and was first implemented on an 11/23 RX02 based system used as a dedicated controller for a random wave generator. Equipped with a servo-controlled hydraulic "paddle" the system produces waves within the flume of a known height and wave period. Typically, at the opposite end of the flume, a dynamically scaled model of a marine structure is placed and its response to the applied sea conditions is measured. The 11/23 is used firstly to drive the hydraulic paddle using a predefined wave spectra and to simultaneously log data from the dynamic model. Using this technique it is possible to subject the model to more realistic "random" sea conditions and so obtain more accurate response data than that available through monochromatic wave testing.

Some 13 separate FORTRAN programs form the basis of the flume control system which is conveniently interlinked by SPECS as shown in Figure 8 using six menu files. In this case only the BASIC command set has been used, primarily with CHAIN and MOVE but with five SEND commands included. A foreground control and logging program is scheduled in this way and analysis routines are chained in the background. A display and plotting menu was conveniently added as the system was developed.

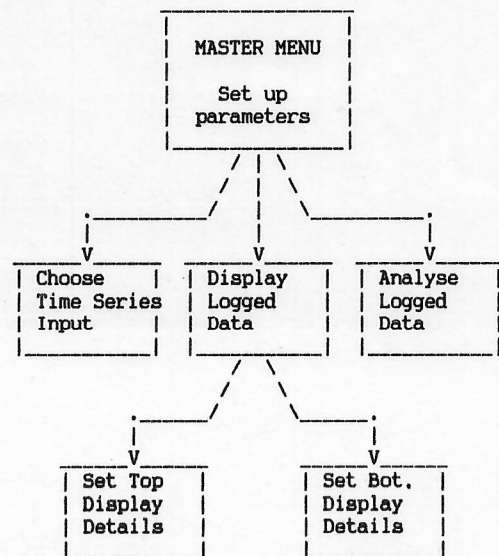


Figure 8 Wave Flume Control System

System Generation Application

As most users know, RT-11 is an operating system rich in options and, together with FORTRAN and its extensions such as MINC, LSP, RGL, IBS and SSP, is a formidable software collection. Combining all this with three system installations with varying hardware configurations, the need to maintain a high level of system reliability, a word processing and data base system thrown in for good measure and there emerges a hazardous and lengthy system generation task.

To alleviate this problem a two stage software generation system was designed with SPECS as the focal point. This system begins where Autopatch finishes and guides the user through the various intricate stages of customisation needed for our systems, which are RX02 based. Figure 9 summarises the second stage of this DISK GEN (2) procedure where various software components are regrouped and packaged into particular system or utility disks according to hardware or application requirements. The total system comprises 45 menus and 36 indirect command files. Each menu generally only contains a single option but contains extensive information regarding the operation to be performed and the disks necessary at various stages (in a similar way to Autopatch). The major commands used include MOVE and SEND with considerable use of BRANCH to allow the single option menus to proceed to the next level of generation after an indirect command file is executed. CHAIN is not used at all - there are no other FORTRAN programs used except SPECS.

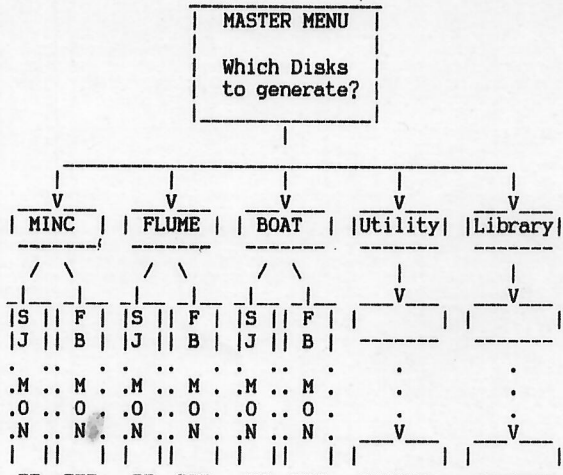


Figure 9 DISK GEN Menu Structure

Boat Computer System

Much of the investigation work carried out at the Hydraulics Laboratory involves the collection of physical data in the field - very often from boats at sea or in coastal rivers and lakes. To greatly improve the quality of data collected over manual means, and to achieve a much higher frequency of readings, an 11/23 based "go anywhere" data logging system has been developed (3). The data to be collected is always of a real time nature and in general consists of a time, a position and one or more related measurements. This means the system must interface to a wide range of instruments such as echo sounders, current meters, temperature/salinity sensors, fluorometers and RADAR systems. Because of this broad range of possible applications the need for an integrated software system from the user point of view is essential, and

SPECS satisfies this requirement. An overview of the menu structure used in a fluorometric dye tracking experiment as part of an ocean outfall investigation is shown in Figure 10.

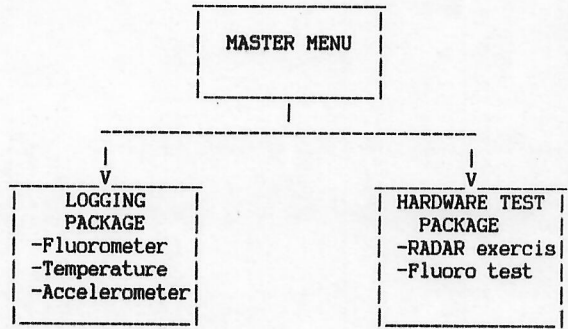


Figure 10 Boat Computer FLULOG Package

The boat computer system has, as its present console, a combined QWERTY membrane keyboard with inbuilt two line liquid crystal display. This has proved an excellent control system for field conditions and together with the SPECS menu truncate feature allows use of exactly the same menu system as used for software development and testing at the Laboratory.

SUMMARY

SPECS is an extremely powerful software tool rich in commands and based on a simple easy-to-use framework of ASCII files. In particular, it is well suited to the control of a number of discrete programs which together define a system or process. It has features designed to aid the inexperienced or less sophisticated user but does not necessarily restrict access to its more advanced or streamlined operations. Because SPECS is itself modular in design the system builder can easily replace, add to, or enhance the command set as required. In addition it achieves this with relatively low overheads, not a single line of MACRO, and without upsetting RT-11. It is planned to submit SPECS to the DECUS library later this year.

ACKNOWLEDGEMENTS

Many thanks to Laboratory colleagues M.Todd, A.Webb and M.Sherringham who have contributed fine ideas to the development of SPECS and for acting as guinea pigs at various stages. This paper is published with the kind permission of the Chief Engineer, Public Works Department of New South Wales.

REFERENCES

1. Public Works Department N.S.W., "User's Guide to SPECS", Manly Hydraulics Laboratory Report, (in preparation).
2. Public Works Department N.S.W., "PDP-11 System Managers Manual", Manly Hydraulics Laboratory Report No. 335, April 1981.
3. Allan, A.J. and Todd, M., "A Mobile Computer System for Data Capture Tasks", Proc. Digital Equipment User's Soc, Vol 7, No.5, Brisbane, 1981, pp.1549-1555.