NAME ar -- archive

SYNOPSIS ar key afile name1 ...

DESCRIPTION ar maintains groups of files combined into a single archive file. Its main use is to create and update library files as used by the loader. It can be used, though, for any similar purpose.

key is one character from the set drtux, optionally concatenated with v. afile is the archive file. The names are constituent files in the archive file. The meanings of the key characters are:

d means delete the named files from the archive file.

r means replace the named files in the archive file. If the archive file does not exist, r will create it. If the named files are not in the archive file, they are appended.

p prints a table of contents of the archive file. If no names are given, all files in the archive are tabled. If names are given, only those files are tabled.

u is similar to r except that only those files that have been modified are replaced. If no names are given, all files in the archive that have been modified will be replaced by the modified version.

x will extract the named files. If no names are given, all files in the archive are extracted. In neither case does x alter the archive file.

v means verbose. Under the verbose option, ar gives a file–by–file description of the making of a new archive file from the old archive and the constituent files. The following abbreviations are used:

c copy
a append
d delete
r replace
x extract

FILES /tmp/vtma, vtmb ... temporary

SEE ALSO ld

DIAGNOSTICS "Bad usage", "afile -- not in archive format", "cannot open temp file", "name -- cannot open", "name -- phase error", "name -- cannot create", "no archive file", "cannot create archive file", "name -- not found".

BUGS

Option l (table with more information) should be implemented.

There should be a way to specify the placement of a new file in an archive. Currently, it is placed at the end.

OWNER

ken, dmr

NAME as -- assembler

SYNOPSIS <u>as</u> $name_1$ . . .

DESCRIPTION

<u>as</u> assembles the concatenation of $name_1$, ... <u>as</u> is based on the DEC–provided assembler PAL–11R [references], although it was coded locally. Therefore, only the differences will be recorded.

Character changes are:

for	use
@	*
#	$
;	/

In <u>as</u>, the character ";" is a logical new line; several operations may appear on one line if separated by ";".
Several new expression operators have been provided:

\>	right shift (logical)
\<	left shift
*	multiplication
\/	division
%	remainder (no longer means "register")
!	one's complement
[]	parentheses for grouping
^	result has value of left, type of right

For example location 0 (relocatable) can be written "0^."; another way to denote register 2 is "2^r0"

All of the preceding operators are binary; if a left operand is missing, it is taken to be 0. The ! operator adds its left operand to the one's complement of its right operand.

There is a conditional assembly operation code different from that of PAL–11R (whose conditionals are not provided):

```
 .if expression
 ...
.end if
```

If the <u>expression</u> evaluates to non–zero the section of code between the ".if" and the .endif is assembled; otherwise it is ignored. ".if" s may be nested.

Temporary labels like those introduced by Knuth [reference] may be employed. A temporary label is defined as follows:

n:

where n is a digit 0 ,.. 9. Symbols of the form "nf" refer to the first label n: following the use of the symbol; those of the form nb refer to the last "n:". The same n may be used many times. Labels of this form are less taxing both on the imagination of the programmer and on the symbol table space of the assembler.

The PAL-11R opcodes ".eot" and ".end" are redundant and are omitted.

The symbols

r0 ... r5
sp
pc
ac
mq
div
mul
lsh
ash
nor
csw

are redefined with appropriate values. The symbol csw refers to the console switches. ".." is the relocation constant and is added to each relocatable symbol; normally it is 40000(8); it may be changed to assemble a section of code at a location different from that in which it will be executed.

It is illegal to assign a value to "."less than its current value.

The new opcode "sys" is used to specify system calls. Names for system calls are predefined. See the section on system calls for their names.

Strings of characters may be assembled in a way more convenient than PAL-11's ".ascii" operation (which is, therefore, omitted). Strings are included between the string quotes '<' and '>' :

<here is a string>

Escape sequences exist to enter non graphic and other difficult characters. These sequences are also effective in single and double character constants introduced by single (') and double (") quotes respectively.

	use for
\n	newline (012)
\0	NULL (000)
\>	>
\t	TAB (011)
\\	\

The binary output of the assembler is placed on the file a.out in the current directory. a.out also contains the symbol table from the assembly and relocation bits. The output of the assembler is executable immediately if the assembly was error—free and if there were no unresolved external references. The link editor ld may be used to combine several assembly outputs and resolve global symbols.

The multiple location counter feature of PALIIR is not supported.

The assembler does not produce a listing of the source program. This is not a serious drawback; the debugger db discussed below is sufficiently powerful to render a printed octal translation of the source unnecessary.

FILES

/etc/as2	pass 2 of the assembler
a.tmp1	temporary
a.tmp2	temporary
a.tmp3	temporary
a.out	object

SEE ALSO ld, nm, sh, un, db, a.out (format of output)

DIAGNOSTICS When an input file cannot be read, its name followed by a question mark is typed and assembly ceases.

When syntactic or semantic errors occur, a single—character diagnostic is typed out together with the line number and the file name in which it occurred. Errors in pass I cause cancellation of pass 2. The possible errors are:

) parentheses error] parentheses error
* Indirection ("*") used illegally
A error in Address
B Branch instruction has too remote an address
E error in Expression
F error in local (or "b") type symbol
G Garbage (unknown) character
M Multiply defined symbol as label
0 Odd—— word quantity assembled at odd

address

P Phase error " .." different in pass 2 from pass 1 value

R Relocation error

U Undefined symbol

X syntaX error

BUGS Symbol table overflow is not checked.

OWNER dmr

NAME B -- language

SYNOPSIS sh rc /usr/b/rc name

DESCRIPTION B is a language suitable for system programming. It is described is a separate publication B reference manual.

The canned shell sequence in /usr/b/rc will compile the program name.b into the executable file a.out. It involves running the B compiler, the B assembler, the assembler and the link editor. The process leaves the files name.i and name.s in the current directory.

FILES name.b, name.i, name.s.

SEE ALSO /etc/bc, /etc/ba, /etc/brt1, /etc/brt2, /etc/bilib, /etc/libb.a, B reference manual.

DIAGNOSTICS see B reference manual

BUGS There should be a B command.

OWNER ken, dmr

NAME bas —— basic

SYNOPSIS bas [file]

DESCRIPTION bas is a dialect of basic. If a file argument is provided, the file is used for input before the console is read.

bas accepts lines of the form:

```
statement
integer statement
```

Integer numbered statements (known as internal statements) are stored for later execution. They are stored in sorted ascending order. Non— numbered statements are immediately executed. The result of an immediate expression statement (that does not have '=' as its highest operator) is printed.

Statements have the following syntax: (expr is short for expression)

expr

The expression is executed for its side effects (assignment or function call) or for printing as described above.

done

Return to system level.

draw expr expr expr

draw is used to draw on a 611—type storage scope through a TSP—i plotter interface. The coordinates of the scope face are zero to one in both the x and y directions. (Zero,zero being the lower left corner.) The expressions are evaluated and design at— ed X, Y, and Z. A line is drawn from the previous X, Y to the new X, Y • If Z is non—zero, the line is visible, otherwise the line is invisible.

for name = expr expr statement
for name = expr expr
... .
next

The for statement repetatively executes a statement (first form) or a group of statements (second form) under control of a named variable. The variable takes on the value of the first expression, then is incremented by one on each loop, not to exceed the value of the second expression.

goto expr

The expression is evaluated, truncated to an integer and execution goes to the corresponding integer numbered statement. If executed from immediate mode, the internal statements are compiled first.

_if_ expr statement

The statement is executed if the expression evaluates to non—zero.

list [expr [expr]]

list is used to print out the stored internal statements. If no arguments are given, all internal statements are printed. If one argument is given, only that internal statement is listed. If two arguments are given, all internal statements inclusively between the arguments are printed.

print expr

The expression is evaluated and printed.

return expr

The expression is evaluated and the result is passed back as the value of a function call.

run

The internal statements are compiled. The symbol table is re—initialized. The random number generator is re—set. Control is passed to the lowest numbered internal statement.

Expressions have the following syntax:

name

A name is used to specify a variable. Names are composed of a letter ('a' — 'z') followed by letters and digits. The first four characters of a name are significant.

number

A number is used to represent a constant value. A number is composed of digits, at most one decimal point ('.') and possibly a scale factor of the form e digits or _e_— digits.

(expr)

Parentheses are used to alter normal order of evaluation.

expr op expr

Common functions of two arguments are

abbreviated by the two arguments separated by an operator denoting the function. A complete list of operators is given below.

expr ([expr [, expr]]
Functions of an arbitrary number of arguments can be called by an expression followed by the arguments in parentheses separated by commas. The expression evaluates to the line number of the entry of the function in the internally stored statements. This causes the internal statements to be compiled. If the expression evaluates negative, an builtin function is called. The list of builtin functions appears below.
name [expr [, expr ...]]
Arrays are not yet implemented.

The following is the list of operators:

= is the assignment operator. The left operand must be a name or an array element. The result is the right operand. Assignment binds right to left, all other operators bind left to right.

& |
& (logical and) has result zero if either of its arguments are zero. It has result one if both its arguments are non—zero. | (logical or) has result zero if both of its arguments are zero. It has result one if -either of its arguments are non—zero.

< <= > >= == <>
The relational operators (< less than, <= less than or equal, > greater than, >= greater than or equal, == equal to, <> not equal to) return one if their arguments are in the specified relation. They return zero otherwise. Relational operators at the same level extend as follows: a>b>c is the same as a>b&b>c.

+ −

Add and subtract.

* /
Multiply and divide.
^

Exponeniation.

-The following is a list of builtin functions:

arg
Arg(i) is the value of the ith actual parameter on the current level of function call.

exp
Exp(x) is the exponential function of x.

log
Log(x) is the logarithm base of x.

sin
Sin(x) is the sine of x (radians).

co s
Cos(x) is the cosine of x (radians).

atn
Atn(x) is the arctangent of x. (Not implemented.)

md
Rnd() is a uniformly distributed random number between zero and one.

expr
Expr() is the only form of program input. A line is read from the input and evaluated as an expression. The resultant value is returned.

int
Int(x) returns x truncated to an integer.

FILES /tmp/btma, btmb •.. temporary

SEE ALSO

DIAGNOSTICS Syntax errors cause the incorrect line to be typed with an underscore where the parse failed. All other diagnostics are self explanatory.

BUGS Arrays [] are not yet implemented. In general, program sizes, recursion, etc are not checked, and cause trouble.

OWNER ken

NAME bcd — binary coded decimal conversion

SYNOPSIS bcd [string]

DESCRIPTION bcd will convert a string into GECOS card code. If no argument string is provided, will read a line and convert it.

FILES

SEE ALSO

DIAGNOSTICS

BUGS

OWNER dmr

NAME boot -- reboot system

SYNOPSIS /etc/boot

DESCRIPTION boot logically a command, and is kept in /etc only to lessen the probability of its being invoked by accident or from curiosity. It reboots the system by jumping to the read—only memory, which contains a disk boot program.

FILES

SEE ALSO boot procedure

DIAGNOSTICS

BUGS Should obviously not be executable by the general user. Also, it should reboot in a more direct manner. The mechanism invoked by jumping to the ROM loader is sensitive to the contents of the console switches, which makes the whole procedure even more dangerous.

Rather than jumping to the ROM, boot should simulate the ROM action with 173700 in the switches. In this manner, It may be used when the switches are not. set, and even in installation without a ROM.

OWNER ken

NAME cat -- concatenate and print

SYNOPSIS _cat_ file1 ...

DESCRIPTION _cat_ reads each file in sequence and writes it on the standard output stream. Thus:

```
cat file
```

is about the easiest way to print a file. Also:

```
cat file1 file2 >file3
```

is about the easiest way to concatenate files.

If no input file is given _cat_ reads from the standard input file.

FILES

SEE ALSO pr, cp

DIAGNOSTICS none; if a file cannot be found it is ignored.

BUGS

OWNER ken, dmr

NAME chdir -- change working directory

SYNOPSIS chdir directory

DESCRIPTION directory becomes the new working directory.

Because a new process is created to execute each command, chdir would be ineffective if it were written as a normal command. It is therefore recognized and executed by the Shell.

FILES

SEE ALSO sh

DIAGNOSTICS ?

BUGS

OWNER ken, dmr

NAME check -- file system consistency check

SYNOPSIS check [filesystem [[blockno_1 ...]]

DESCRIPTION check will examine a file system, build a bit map of used blocks, and compare this bit map against the bit map maintained on the file system. If the file system is not specified, a check of both /dev/rf0 and /dev/rk0 is performed. Output includes the number of files on the file system, the number of these that are 'large', the number of used blocks, and the number of free blocks.

FILES /dev/rf0, /dev/rk0

SEE ALSO find

DIAGNOSTICS Diagnostics are produced for blocks missing, duplicated, and bad block addresses. Diagnostics are also produced for block numbers passed as parameters. In each case, the block number, i—number, and block class (i = inode, x indirect, f free) is printed.

BUGS The checking process is two pass in nature. If checking is done on an active file system, extraneous diagnostics may occur.

The swap space on the RF file system is not accounted for and will therefore show up as 'missing'.

OWNER ken, dmr

NAME chmod -- change mode

SYNOPSIS chmod octal file

DESCRIPTION The octal mode replaces the mode of each of the files. The mode is constructed from the OR of the following modes:

01 write for non—owner
02 read for non—owner
04 write for owner
10 read for owner
20 executable
40 set—UID

Only the owner of a file may change its mode.

FILES

SEE ALSO stat, ls

DIAGNOSTICS

BUGS

OWNER ken, dmr

NAME chown -- change owner

SYNOPSIS chown owner file

DESCRIPTION owner becomes the new owner of the files. The owner may be either a decimal UID or a name found in /etc/uids.

Only the owner of a file is allowed to change the owner. It is illegal to change the owner of a file with the set—user—ID mode.

FILES /etc/uids

SEE ALSO stat

DIAGNOSTICS

BUGS

OWNER ken, dmr

NAME	cmp -- compare two files
SYNOPSIS	cmp file1 file2
DESCRIPTION	The two files are compared for identical contents. Discrepancies are noted by giving the offset and the differing words.
FILES	
SEE ALSO	
DIAGNOSTICS	Messages are given for inability to open either argument, premature EOF on either argument, and incorrect usage.
BUGS	If the two files differ in length by one byte, the extra byte does not enter into the comparison.
OWNER	dmr

NAME cp -- copy

SYNOPSIS cp file12 file12 file21 file22 ...

DESCRIPTION Files are taken in pairs; the first is opened for reading, the second created mode 17. Then the first is copied into the second.

FILES

SEE ALSO cat, pr

DIAGNOSTICS Error returns are checked at every system call, and appropriate diagnostics are produced.

BUGS The second file should be created in the mode of the first.

A directory convention as used in _mv_ should be adopted to cp.

OWNER ken, dmr

NAME	date -- print the date
SYNOPSIS	date
DESCRIPTION	The current date is printed to the second.
FILES	
SEE ALSO	sdate
DIAGNOSTICS	
BUGS	
OWNER	dmr

NAME db -- debug

SYNOPSIS db [core [namelist]]

DESCRIPTION Unlike many debugging packages (including DEC's ODT, on which db is loosely based) db is not loaded as part of the core image which it is used to examine; instead it examines files. Typically, the file will be either a core image produced after a fault or the binary output of the assembler. Core is the file being debugged; if omitted "core" is assumed. namelist is a file containing a symbol table. If it is omitted, a.out is the default. If no appropriate name list file can be found, db can still be used but some of its symbolic facilities become unavailable.

The format for most db requests is an address followed by a one character command.

Addresses are expressions built up as follows:

1. A name has the value assigned to it when the input file was assembled. It may be relocatable or not depending on the use of the name during the assembly.

2. An octal number is an absolute quantity with the appropriate value.

3. An octal number immediately followed by "r" is a relocatable quantity with the appropriate value.

4. The symbol "." indicates the current pointer of db. The current pointer is set by many requests.

5. Expressions separated by "+" or " "(blank) are expressions with value equal to the sum of the components. At most one of the components may be relocatable.

6. Expressions separated by "−" form an expression with value equal to the difference to the components. If the right component is relocatable, the left component must be relocatable.

7. Expressions are evaluated left to right. Names for registers are built in:

```
r0 ... r5
   sp
   pc
```

ac
mq

These may be examined. Their values are deduced from the contents of the stack in a core image file. They are meaningless in a file that is not a core image.

If no address is given for a command, the current address (also specified by ".") is assumed. In general, . points to the last word or byte printed by ~.

There are db commands for examining locations interpreted as octal numbers, machine instructions, ASCII characters, and addresses. For numbers and characters, either bytes or words may be examined. The following commands are used to examine the specified file.

/ The addressed word is printed in octal.

\ The addressed byte is printed in octal.

" The addressed word is printed as two ASCII characters.

' The addressed byte is printed as an ASCII character.

` The addressed word is multiplied by 2, then printed in octal (used with B programs, whose addresses are word addresses).

? The addressed word is interpreted as a machine instruction and a symbolic form of the instruction, including symbolic addresses, is printed. Usually, the result will appear exactly as it was written in the source program.

& The addressed word is interpreted as a symbolic address and is printed as the name of the symbol whose value is closest to the addressed word, possibly followed by a signed offset.

<nl> (i. e., the character "new line") This command advances the current location counter • and prints the resulting location in the mode last specified by one of the above requests.

This character decrements "." and prints the resulting location in the mode last selected one of the above requests. It is a converse to <nl>.

It is illegal for the word—oriented commands to have odd addresses. The incrementing and decrementing of "." done by the <nl> and requests is by one or two depending on whether the last command was word or byte oriented.

The address portion of any of the above commands may be followed by a comma and then by an expression. In this case that number of sequential words or bytes specified by the expression is printed. "."is advanced so that it points at the last thing printed.

There are two commands to interpret the value of expressions.

= When preceded by an expression, the value of the expression is typed in octal. When not preceded by an expression, the value of. "." is indicated. This command does not change the value of ".".

: An attempt is made to print the given expression as a symbolic address. If the expression is relocatable, that symbol is found whose value is nearest that of the expression, and the symbol is typed, followed by a sign and the appropriate offset. If the value of the expression is absolute, a symbol with exactly the indicated value is sought and printed if found; if no matching symbol is discovered, the octal value of the expression is given.

The following command may be used to patch the file being debugged.

! This command must be preceded by an expression. The value of the expression is stored at the location addressed by the current value of "." . The opcodes do not appear in the symbol table, so the user must assemble them by hand.

The following command is used after a fault has caused a core image file to be produced.

$ causes the contents of the general registers and several other registers to be printed both in octal and symbolic format. The values are as they were at the time of the fault.

The only way to exit from _db_ is to generate an end of file on the typewriter (EOT character).

FILES

SEE ALSO as; core for format of core image.

"

DIAGNOSTICS "File not found" the first argument cannot be read; otherwise "?"

BUGS Really, db should know about relocation bits, floating point operations, and PDP11/45 instructions.

OWNER dmr

NAME dbppt -- dump binary paper tape

SYNOPSIS _dbppt_ name [output]

DESCRIPTION _dbppt_ produces binary paper tape in UNIX standard format, which includes checksums and a zero—suppression feature. File name is dumped; if the output argument is not given, output goes to /dev/ppt.

FILES /dev/ppt

SEE ALSO _lbppt_ to reload the tapes. bppt for binary paper tape format.

DIAGNOSTICS ?

BUGS

OWNER ken