ALCOR C

The Alcor C Programming System Overview

- Alcor C is a complete program development system for the C language. All the tools necessary to edit, compile, and execute C programs are provided.
- The Alcor C compiler is a complete implementation of the C language. It contains all the features described in the Kernighan and Ritchie book, "The C Programming Language," and provides many additional features as well.

Easy to Use

- Creating C source files is made easy by a powerful full screen text editor. The editor is user configurable and contains macro command capability.
- Compiling a C source file is made easy by a fast one pass compiler. No preprocessing steps are required and no intermediate files are generated.
- The compiler generated object code may be directly executed with a special runtime utility. No time consuming linking is necessary.

Compact

- The Alcor C compiler generates an extremely compact object code. This allows for the development of very large C programs.
- For even more compactness, a peephole optimizer is provided which will reduce object code size by an additional 10% to 30%.
- The Alcor linkage editor may be used to link minimal runtime support to the object code. This makes it possible for programs to include only the runtime necessary for the specific language features used.

Fast

- The object code generated by the Alcor C compiler is very efficient. For most applications, program execution speed is more than adequate.
- For even more speed, a code generator is provided for translating object code to native processor code.
- The code generator may also be used to generate assembly language. Critical optimizations may be performed to obtain the fastest possible execution speed. The assembly language may then be asembled by the Alcor assembler.

Versatile

- The Alcor C support utilities allow you to customize each C program according to size and speed requirements.
- The compact object code generated by the compiler may be used to obtain the smallest size. The object code may be translated to pure processor code to obtain the fastest execution speed. Object code and processor code may be mixed to utilize the advantages of both.

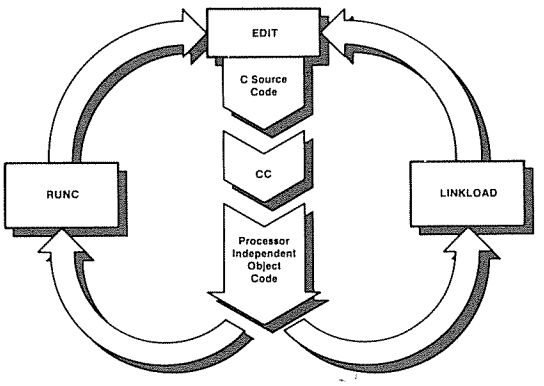
Integrated

- The design of Alcor C is fully compatible with the Alcor Pascal and Alcor Multi-Basic languages.
- A program written in Alcor C may use libraries written in either Alcor Pascal or Alcor Multi-Basic.
- The Alcor support tools are also compatible among all three languages.

Portable

- The portability of the C language is greatly enhanced by the design of Alcor C and its availability on a wide range of computers and operating systems.
- Not only is the C source code portable, the compiler generated object code is portable as well. A single computer system may be used to develop programs that are targeted for a wide range of other computer systems.

Program Development with Alcor C (Fast Development Cycle)



EDIT - The editor is used to create the C source file.
CC - The C compiler translates the C source file, creating an object file.
RUNC - The C run utility loads and executes the object file.
LINKLOAD - The linking loader loads the object file. Other object files may be loaded and linked together. The object code may then be executed or an executable command file may be created.

C Language Features

Alcor C is a complete implementation of the C language as described by Kernighan and Ritchie.

• Macro Preprocessor

Included in the single compile pass is a macro preprocessor which supports parameter substitution. The following macro commands are supported.

#include	#if	#ifndef
#define	#endif	#else
#undef	#ifdef	#line

• Data Types

type	length	range
char	8 bits	0 to 255
int	16 bits	-32768 to 32767
short	16 bits	-32768 to 32767
unsigned	16 bits	0 to 65535
pointer	16 bits	0 to 0xFFFF
long	32 bits	-2147483648 to 2147483647
float	32 bits	(-)1.7E-38 to (-)1.7E+38 6 digits accuracy
double	64 bits	(-)1.7E-38 to (-)1.7E+38 16 digits accuracy

• Operators

All operators, including sizeof and cast.

• Storage Classes

extern static auto register

• Other Features Supported

typedef	struct (including bit fields)
union	initializers

• Complete I/O Support

Contains a complete library of Unix compatible support routines.

printf	fprintf	getc	fgets
scanf	fscanf	putc	fputs
sprintf	getchar	gets	fopen
sscanf	putchar	puts	fclose

Command line I/O redirection is supported.

• Standard C Functions

isalpha	isspace	toupper	strcpy
isdigit	tolower	strlen	strcmp
atoi	atof	calloc	cfree
exit	_exit		

• Extended String Handling

Contains a complete library of extended dynamic string handling routines.

len	encodei	decoded	delete
left$	encoder	character	find
right$	encoded	cmpstr	insert
mid$	decodei	conc	replace
str$	decoder	cpystr	

• Powerful Library Routines

Contains an extensive library of routines to extend the power of C.

openrand	io$error	writestring
readrand	file$status	writech
writerand	getkey	out
closerand	inkey	inp
call$	$memory	setacnm
hp$error	user	

• Mathematical Functions

Included is a mathematics package to enhance the math capability of the C language. Both single and double precision are supported.

cos	sqr	sqrt
ln	sin	abs
exp	arctan	

Programs Included

The Alcor C Programming System contains the following development tools.

EDIT
A powerful full screen text editor which is user configurable. Editor commands may be mapped to keys as desired using the powerful macro command capability. (the editor supplied with 48K systems is not user configurable)

CC
The C compiler which translates C source programs to efficient processor independent object code.

RUNC
The run utility which directly executes the compiler generated object code.

LINKLOAD
The linking loader which will link seperately compiled program segments and build executable command files.

OPTIMIZE
The object code optimizer which reduces object code size by 10% to 30%.

CODEGEN
The code generator which translates the compiler generated object code to native processor code for increased execution speed.

EXTENSIVE DOCUMENTATION

Beginners Guide
Introduces you to the C programming system.

System Implementation Manual
Explains the processor and operating system dependent features of the system.

Editor Manual
A detailed description of the full screen text editor.

C Tutorial
An introduction to programming with the C language.

C Reference Manual
A detailed description of the Alcor C language.

System Requirements

CPU	Z80, 8086 or 8088
Disk Drives	2 Recommended
Operating system	MSDOS 2.x, PCDOS 2.x CP/M 2.2, CP/M 3.0 TRSDOS, LDOS NEWDOS, DOSPLUS
Computers supported	IBM PC, PC Jr. Tandy 1000, 1200, 2000 TRS80 Model I, III, 4 Most Z80 CP/M systems including: Kaypro, Osborne, Morrow, NEC, Epson etc.
Memory required	MS/PCDOS - 96k CP/M - 52k TRSDOS - 48k

ALCOR Systems
1132 Commerce Dr.
Richardson, Texas 75081
(214) 238-8554

MULTI-BASIC

Multi-Basic Provides

- Speed
- Power
- Compatiblity
- Portability
- Versatility

Ideal for the Beginner

The interactive environment of interpreted BASIC makes learning to program easier. With Multi-Basic, you can obtain the speed and power of a compiled language without giving up the advantages of interactive debugging.

- Multi-Basic can compile the programs written for interpreted Basic with few or no changes. Existing Basic programs run 1 to 50 times faster after being compiled by Multi-Basic.
- You may continue to use interpreted Basic to debug programs and then use Multi-Basic to obtain a big improvement in performance.
- Mulit-Basic adds advanced features to the Basic language. Once you have learned the fundamentals of programming, you can go on to the more powerful features that Multi-Basic provides.
- You can learn structured and modular programming skills with Multi-Basic.
- When you move up to a new computer, your programs will move with you. The availability of Multi-Basic on a wide range of computers means that you won't be spending valuable time rewriting programs.

A Must for the Professional

Multi-Basic provides the professional programmer with the most powerful and versatile Basic language system ever developed.

- Advanced features not present in any other Basic allow the use of programming techniques previously supported only by languages such as Pascal.
- Large programs may be developed.
- Separate compilation allows you to build libraries of general purpose routines which may be linked to any programs that need them.
- Compatibility with Alcor Pascal and Alcor C allows Multi-Basic programs to use routines written in Pascal or C.
- Advanced support tools allow you to optimize each program based on size and speed requirements.
- Executable command files may be created and distributed with no runtime licensing fees.
- A single computer system may be used to develop software which will execute on a wide range of other computers.

EXTENDED BASIC FEATURES

Multi-Basic provides the extra features which transform Basic into a professional development language.

- **Multi-Line Functions**

Modular program development is possible; with functions that allow multiple lines of statements. This provides the ability to create libraries of routines which are used by many different programs.

- **Local Variables**

Variables declared inside a function are local to the function. Local variable names may be chosen freely without conflicting with the variable names used elsewhere in the program.

- **Structured Statements**

The WHILE statement and the IF THEN ELSE statement have been added to promote structured programming.

- **Statement Labels**

Descriptive statement labels may be used in place of line numbers for more readable programs.

- **Long Variable Names**

Variable names may be up to 255 characters long. Descriptive variable names make a program easier to maintain.

- **Dynamically Dimensioned Arrays**

The size of an array can be changed while a program is executing. The memory used by an array is recovered when its size is changed.

- **Dynamic Memory Allocation**

Multi-Basic provides dynamic allocation of local variables. The local variables of a function occupy memory only while the function is executing. This makes efficient use of memory and allows larger programs to be written.

- **Nested Functions**

Multi-Basic supports nested functions (the definition of a function within a function). This feature aids modular programming by allowing functions to be broken into smaller functions.

- **Recursion**

Multi-Basic allows recursive programming. Recursion is the ability of a function to call itself. Many programming problems are simplified through the use of recursion.

- **Complete Function Library**

A function library is provided which includes the functions supported by interpreted and commercial Basic.

SOLVING THE PORTABILITY PROBLEM

A fundamental problem with Basic has been the lack of portability. Interpreted Basic is available on a large number of computers but each version is incompatible with the next. Commercial Basic is not available for the TRS-80 line of computers.

Alcor Multi-Basic provides a solution to Basic's portability problem. Using Multi-Basic, interpreted Basic programs

Extending Interpreted Basic

By using only the features supported by interpreted Basic, you can benefit from an interactive environment when testing your programs. The completed program can then be compiled by Multi-Basic to obtain better performance. Moreover, with some simple programming techniques, the power of interpreted Basic can be extended to take advantage of the multi-line function capability of Multi-Basic.

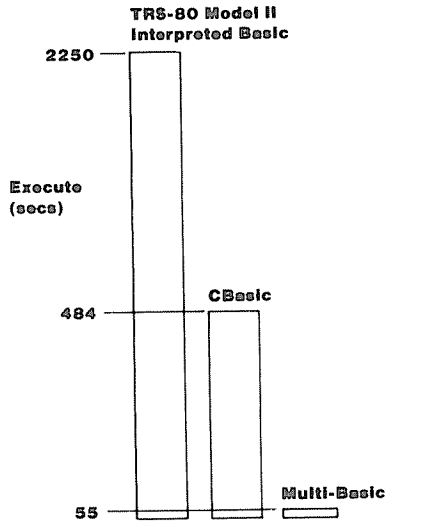
For example, suppose you create a library of graphics routines for plotting. You can interactively test a program which uses the graphics routines by using PRINT statements in place of calls to the plot routines. Once the program is debugged, the PRINT statements are replaced by the actual function calls and the program is compiled.

VERSATILITY OF ALCOR MULTI-BASIC

The Multi-Basic system may be used in many ways.

Process Existing Programs

The Multi-Basic compiler can compile existing programs written in interpreted Basic or CBasic to provide increased execution speed. The following diagram illustrates the speed improvement.



Execution times for the benchmark program published in Sept/81 Byte Magazine

Using Advanded Features

Multi-Basic provides the most power when you take advantage of all the advanced language features it offers. Line numbers may be replaced by descriptive statement labels. Liberal use of structured statements in place of gotos makes programs easier to read. Modular programming using multi-line functions makes programs easier to maintain.

ALCOR MULTI-BASIC PROGRAMMING SYSTEM

Alcor Multi-Basic is a complete program development system that includes all the tools necessary to create, compile, and execute Basic programs.

CREATING PROGRAMS

A powerful full screen text editor makes it easy to create Basic programs. On systems with 52K or more of memory, the supplied editor is user configurable. Editor commands may be mapped to keys as desired. You can create your own commands with the macro command capability.

COMPILING PROGRAMS

Compiling a program is simple with the fast, one pass Multi-Basic compiler.

MULTIBAS filename

This command causes the compiler to translate the Basic program in the specified file to an executable object code.

EXECUTING PROGRAMS

The object code generated by the compiler may be directly executed by a special utility.

RUNB filename

This command causes the object code in the specified file to be executed.

A linking loader provides the ability to link separately compiled program segments into a single program. The LINKLOAD utility also allows you to create

OPTIONAL PROGRAM OPTIMIZATION

The Multi-Basic compiler generates an extremely compact and fast executing object code which is quite adequate for most programs. However, for very large or speed critical programs, two utilities are provided for customization.

The OPTIMIZE utility is an object code optimizer which reduces the size of object code by an additional 10 to 30 percent.

The CODEGEN utility is a code generator which translates object code to native processor instructions. This provides a 3 to 5 times increase in execution speed.

Object code and native processor instructions may be mixed to combine the size advantage of object code with the speed advantage of processor instructions. No other language system offers this flexibility.

System Requirements

CPU	Z80, 8086 or 8088
Disk Drives	2 Recommended
Operating system	MSDOS 2.x, PCDOS 2.x CP/M 2.2, CP/M 3.0 TRSDOS, LDOS NEWDOS, DOSPLUS
Computers supported	IBM PC, PC Jr. Tandy 1000, 1200, 2000 TRS80 Model I, III, 4 Most Z80 CP/M systems including: Kaypro, Osborne, Morrow, NEC, Epson etc.
Memory required	MS/PCDOS - 96k CP/M - 52k TRSDOS - 48k

ALCOR Systems
1132 Commerce Dr.
Richardson, Texas 75081
(214) 238-8554

Multiprocessor Assembler

The Multiprocessor assembler is one of the most versatile assemblers available. It is a full featured assembler that generates relocatable object code that is compatible with Alcor's high level languages. Not only does it assemble Z80 instructions and Alcor's pcode, but it also performs as a cross assembler for several other microprocessors.

With the multiprocessor assembler you can create assembly language procedures and functions that can be directly linked to and called from high level languages. They appear to the caller as if they were written in the high level language. Parameters may be passed and results may be returned. The assembly language function can be called by name rather than address.

Assembler features

- Assembles code for:
  8080
  Z80 (extended 8080 mnemonics)
  1802
  6502
  Pcode
- Assembly language for different processors may be freely mixed in a single program.
- The generated object code is relocatable and compatible with Alcor's high level languages.
- A utility is included to convert an object file into a command file.
- Labels may be up to 8 characters in upper case, lower case or mixed. Case is significant.
- Labels may contain special characters such as "$" and "_".
- Labels may be terminated by a colon or blank.
- Opcodes may be in either upper case or lower case, case is ignored.
- Opcode names may be changed by the user.
- Labels can have the same name as an opcode.
- Comment lines may start with a semicolon (;) or asterisk (*). Blank lines are also comments.

Expressions

- Expressions of any complexity are accepted. There is no limit to the number of operators.
- Arithmetic operators
  "+" addition
  "-" subtraction
  "*" multiplication
  "/" division
- Bitwise boolean operators
  "&" - and
  "!" - or
  "~" - not
- Operators have precedence. Multiplication and division are performed first. Order of operations may be changed using parentheses.
- Constants may be expressed in binary (base 2), decimal (base 10), hexadecimal (base 16) or octal (base 8) or ascii characters (i.e 'p').
- "$" represents the current location.

Listing

- Generated code may be listed in hexadecimal or octal.
- The listing may be turned on and off with the pseudo-ops "LIST" and "NOLIST"
- The listing contains an alphabetized list of defined labels
- A cross reference that identifies each use of each label may included if desired.

Pseudo-operators

- EQU and SET allow constants and variables to be defined.
- ORG specifies the origin of the code.
- DB and DW allow data to be assembled and accept a list of bytes or words. DB also allows string constants.
- DEF defines a symbol and makes it available to other modules.
- REF and RREF allow reference to symbols defined in other modules.
- GLOBAL allows global symbols to be defined for multiple modules that are assembled together.
- LIST, NOLIST, XREF, PAGESIZE, WIDTH, LISTIF and LISTHEX control the format of the listing.
- INCLUDE allows the contents of a file to be included in the assembly. The included file may also contain INCLUDE directives.
- IF and ENDIF may be used for conditional assembly.

Manual

- 93 pages with table of contents and index
- Includes description of the pcode instruction set.
- Describes interfacing to high level languages and includes function and procedure templates.

Advanced Development Package

The advanced development package is the perfect companion to Alcor's high level language compilers. It consists of two utilities, a pcode optimizer (OPTIMIZE) and a native code generator (CODEGEN). The optimizer is used to reduce the size of a high level language program so that larger programs can be executed in the limited memory available in the computer. The native code generator converts pcode to instructions that are directly executed by the microprocessor. This increases the speed of the program.

OPTIMIZE

The primary purpose of the pcode optimizer is to reduce the size of object programs. It does this by replacing instructions and sequences of instructions by shorter instructions or shorter sequences. The function performed by the program is not changed but the size is reduced.

You need to use the optimizer either when the program becomes too large to fit into memory or when it needs more space to use for data. The optimizer also makes some improvements that result in faster execution. For example, expressions involving constants are computed by the optimizer and the resulting program uses the result rather than computing it.

Optimization typically reduces the size of the program by 5% to 30% . The amount of space saved depends on the program. For example, the size of string constants cannot be reduced.

CODEGEN

The native code generator converts the pcode that is produced by the high level language compiler into machine instructions. Since these instructions are executed directly by the microprocessor, the program will execute faster.

CODEGEN produces object code directly, so no assembly step is required. It can also create an assembly language source file if you wish. Functions that are processed by CODEGEN will typically run 3 to 5 times faster than pcode. Some operations, such as input and output and floating point operations do not improve as much. This is because they use subroutines that are already written in assembly language.

The native code produced by CODEGEN can be mixed with pcode. This allows you to improve the speed of the critical functions without a large increase in the size of the program as a whole.

Blaise I
full screen text editor

Version 2.1 of the Blaise I text editor is a much improved version of the original Blaise I. It is much faster and has additional features including the ability to manipulate blocks of text. It allows you to edit as many files as you wish without exiting the editor. When you exit, the file is saved, and you have the option of returning to the operating system or editing another file.

Blaise I features

- Cursor movement ↑ ↓ → ←
  beginning of line, end of line,
  beginning of file, end of file,
  tab, back tab, scroll forward,
  scroll backward, home
- Character insertion and deletion
- Insert line, insert 16 lines
- delete line, undelete line
  delete block
- clear to end of line, duplicate to
  end of line
- Mark a block of text, Jump to mark,
  exchange mark & cursor
- Split line, merge lines
- Set tab at cursor, clear tab at cursor
- Set tabs at regular intervals or
  at specific columns
- Automatic indentation
- Find and replace string
  Find and replace next
- Print any portion of the buffer
- Generates special characters
  [,], ^, {, }, |, _, \, ~, `
- Help
- Show line, move by lines
- Extract a block of text to a file
  Insert a file or portion of a file
- Show file - displays the contents of
  any file on the screen without leaving
  the editor.
- Horizontal scrolling - allows editing of
  files with lines longer than 64 characters.
- Any number of files can be edited
  without leaving the editor

SUPER SPRING SALE

	LIST	SALE
☐ Pascal with editor and ADP	$139.00	89.95
☐ C with editor and ADP	139.00	89.95
☐ Multi-Basic with editor and ADP	139.00	89.95
☐ Blaise I (full screen text editor for TRSDOS 2.3 & 1.3)	49.00	39.00
☐ Blaise II (programmable full screen text editor)	79.00	49.00
☐ Multiprocessor Assembler (XASM)	69.00	✓ 49.00
☐ Pascal Cross Reference (XREF)	29.00	
☐ "The C Programming Language" by Kernighan & Ritchie	18.00	

PLEASE SPECIFY OPERATING SYSTEM

☐ CP/M 2.2
☐ CP/M Plus
☐ TRSDOS (2.3) (for model I)*
☐ TRSDOS (1.3) (for model III)*
☐ TRSDOS (6.X) (for model 4)*
☐ MSDOS (2.0)
☐ PCDOS (1.1)
☐ PCDOS (2.0 & 2.1)

*Pascal for TRSDOS 1.3, 2.3, & 6.X sold exclusively by Radio Shack
C for TRSDOS 6.X sold exclusively by Radio Shack

Available CP/M Formats:

☐ 8" SSSD
☐ Apple II
☐ Kaypro 2
☐ Kaypro 4
☐ Osborne I SD
☐ Osborne I DD
☐ Televideo
☐ Lobo Max 80
☐ Epson QX-10
☐ Xerox 820-1
☐ Morrow Micro Decision
☐ DEC VT-180 (Rainbow)

Available MSDOS Formats:

☐ IBM PC
☐ Tandy 2000

☐ Other _____
(other formats available upon request)

SHIPPING AND HANDLING

For Pascal, C, or Multi-Basic: Add $6 U.S.A. or $28 Overseas.
All other products: Add $3 U.S.A. or $15 Overseas shipping.

Name _____ Company _____
Street _____ Phone # _____
City _____ State _____ Zip _____
☐ MasterCharge/Visa # _____ Exp. Date _____ ☐ check or money order

Price subject to change without notice

ALCOR Systems

1132 Commerce Dr.
Richardson, TX 75081
(214) 238-8554

Alcor Systems
1132 Commerce Dr.
Richardson, Tx. 75081
(214) 238-8554