Whitesmiths, Ltd.

C INTERFACE MANUAL FOR 8086

Edition: 2.2

Date: March 1983

The C language was developed at Bell Laboratories by Dennis Ritchie; Whitesmiths, Ltd. has endeavored to remain as faithful as possible to his language specification. The external specifications of the Idris operating system, and of most of its utilities, are based heavily on those of UNIX, which was also developed at Bell Laboratories by Dennis Ritchie and Ken Thompson. Whitesmiths, Ltd. gratefully acknowledges the parentage of many of the concepts we have commercialized, and we thank Western Electric Co. for waiving patent licensing fees for use of the UNIX protection mechanism.

The successful implementation of Whitesmiths′ compilers, operating systems, and utilities, however, is entirely the work of our programming staff and allied consultants.

For the record, UNIX is a trademark of Bell Laboratories; IAS, RSTS/E, VAX, VMS, P/OS, PDP-11, RT-11, RSX-11M, and nearly every other term with an 11 in it all are trademarks of Digital Equipment Corporation; CP/M is a trademark of Digital Research Co.; MC68000 and VERSAdos are trademarks of Motorola Inc.; ISIS and iRMX are trademarks of Intel Corporation; A-Natural and Idris are trademarks of Whitesmiths, Ltd. C is not.

Copyright (c) 1978, 1979, 1980, 1981, 1982, 1983

by Whitesmiths, Ltd.

All rights reserved.

C INTERFACE MANUAL FOR 8086

SECTIONS

I.	Idris Assembler Conventions for 8086
II.	Programming Utilities
III.a.	Idris System Interface Library
III.b.	CP/M-86 - DOS System Interface Library
III.c.	UDI System Interface Library
IV.	Machine Support Library for 8086

SCOPE

This manual describes the 8086 dependent aspects of the C programming environment provided by Whitesmiths, Ltd. In addition, it documents all of the utilities necessary for building new programs. Section I introduces the conventions and describes the format used by the Idris assembler. Section II succinctly describes the programming utilities of Idris, which also serve as cross support utilities for other host machines. Each subsection of Section III describes the library functions that interface the portable C library to Idris, to CP/M-86 or PCDOS/MSDOS, or to the Intel UDI; and Section IV describes the runtime routines called upon by code produced by the 8086 C compiler.

Information on the C language and the portable library may be found in the C Programmers' Manual, while information peculiar to other machines supported by Whitesmiths, Ltd. is given in other C Interface Manuals.

THIS MANUAL IS PROVIDED WITH SEVERAL SOFTWARE PACKAGES, SOME OF WHICH USE ONLY A SUBSET OF THE FACILITIES DOCUMENTED. THE PRESENCE OF A MANUAL PAGE HERE DOES NOT IMPLY THAT THE CORRESPONDING SOFTWARE IS ALSO SUPPLIED.

TABLE OF CONTENTS

I. Idris Assembler Conventions for 8086

I - 1 As.86 The 8086 Assembly Language

II. Programming Utilities

II - 1 Introduction the programming utilities
II - 2 Conventions using the utilities
II - 8 ROM writing read-only code
II - 10 as.86 assembler for 8086
II - 12 c multi-pass command driver
II - 15 db binary file editor/debugger
II - 21 hex translate object file to ASCII formats
II - 24 lib maintain libraries
II - 27 link combine object files
II - 31 lord order libraries
II - 33 p1 parse C programs
II - 35 p2.86 generate code for Intel 8086 C programs
II - 37 pp preprocess defines and includes
II - 39 prof produce execution profile
II - 41 ptc Pascal to C translator
II - 43 rel examine object files

III.a. Idris System Interface Library

III.a - 1 Interface to Idris system
III.a - 3 Conventions Idris system subroutines
III.a - 5 c86 compile and link C programs
III.a - 6 pc86 compile and link Pascal programs
III.a - 7 Crt C runtime entry
III.a - 8 Crtp set up profiling at runtime
III.a - 10 _pname program name
III.a - 11 close close a file
III.a - 12 create open an empty instance of a file
III.a - 13 exit terminate program execution
III.a - 14 lseek set file read/write pointer
III.a - 15 onexit call function on program exit

III.a - 16	**onintr**	capture interrupts
III.a - 17	**open**	open a file
III.a - 18	**read**	read from a file
III.a - 19	**remove**	remove a file
III.a - 20	**sbreak**	set system break
III.a - 21	**uname**	create a unique file name
III.a - 22	**write**	write to a file
III.a - 23	**xecl**	execute a file with argument list
III.a - 24	**xecv**	execute a file with argument vector

III.b. CP/M-86 - DOS System Interface Library

III.b - 1	**Interface**	to CP/M-86 or DOS system
III.b - 3	**Conventions**	CP/M system subroutines
III.b - 5	**Conventions**	DOS system subroutines
III.b - 7	**c**	compiling C programs
III.b - 8	**pc**	compiling Pascal programs
III.b - 9	**ld**	linking a C program
III.b - 10	**to86**	convert to CP/M-86 or DOS executable format
III.b - 13	**chdr**	C runtime entry for CP/M-86
III.b - 14	**doshdr**	C runtime entry for DOS
III.b - 15	**DOS2**	enhancements for DOS Version 2.0
III.b - 17	**_main**	setup for main call
III.b - 18	**_pname**	program name
III.b - 19	**close**	close a file
III.b - 20	**cpm**	call CP/M-86 system
III.b - 21	**cpmx**	call CP/M-86 system with doubleword result
III.b - 22	**create**	open an empty instance of a file
III.b - 23	**dos**	call DOS system
III.b - 24	**dosx**	call DOS system with doubleword result
III.b - 25	**dosy**	call DOS 2.0 system
III.b - 26	**exit**	terminate program execution
III.b - 27	**lseek**	set file read/write pointer
III.b - 28	**onexit**	call function on program exit
III.b - 29	**onintr**	capture interrupts
III.b - 30	**open**	open an existing file
III.b - 31	**read**	read characters from a file
III.b - 32	**remove**	remove a file
III.b - 33	**sbreak**	set system break
III.b - 34	**uname**	create a unique file name
III.b - 35	**write**	write characters to a file

III.c. UDI System Interface Library

III.c - 1	**Interface**	to UDI systems
III.c - 3	**Conventions**	UDI system subroutines

III.c - 4	c	compiling C programs
III.c - 5	pc	compiling Pascal programs
III.c - 6	clink	linking C programs
III.c - 7	chdr	C runtime entry
III.c - 8	_main	setup for main call
III.c - 9	_pname	program name
III.c - 10	close	close a file
III.c - 11	create	open an empty instance of a file
III.c - 12	exit	terminate program execution
III.c - 13	lseek	set file read/write pointer
III.c - 14	onexit	call function on program exit
III.c - 15	onintr	capture interrupts
III.c - 16	open	open a file
III.c - 17	read	read from a file
III.c - 18	remove	remove a file
III.c - 19	sbreak	set system break
III.c - 20	uname	create a unique file name
III.c - 21	write	write to a file

IV. Machine Support Library for 8086

IV - 1	Conventions	the 8086 runtime library
IV - 2	cs	jump far or read code segment register
IV - 3	ds	load or read data segment register
IV - 4	es	load or read extra segment register
IV - 5	in	input byte from port
IV - 6	inw	input word from port
IV - 7	movs	move memory to memory
IV - 8	out	output byte to port
IV - 9	outw	output word to port
IV - 10	ss	load or read stack segment register
IV - 11	c_cfcc	copy 8087 condition code
IV - 12	c_count	counter for profiler
IV - 13	c_dadd	add double into double
IV - 14	c_dcmp	compare two doubles
IV - 15	c_ddiv	divide double into double
IV - 16	c_dmul	multiply double into double
IV - 17	c_dsub	subtract double from double
IV - 18	c_dtf	convert double to float
IV - 19	c_dtl	convert double to long
IV - 20	c_fac	the floating accumulators
IV - 21	c_fret	return from a far C function
IV - 22	c_ftd	convert float to double
IV - 23	c_iax	jump indirect on ax
IV - 24	c_lcmp	compare two signed longs
IV - 25	c_ldiv	divide signed long by long
IV - 26	c_llsh	shift long left by count
IV - 27	c_lmod	remainder signed long by long
IV - 28	c_lmul	multiply long by long
IV - 29	c_lrsh	shift signed long right by count

IV - 30	c_ltd	convert signed long to double
IV - 31	c_repk	repack a double number
IV - 32	c_ret	return from a C function
IV - 33	c_rets	return from a C function, restoring no registers
IV - 34	c_sav	save register on entering a C function
IV - 35	c_savs	save register on entering a C function
IV - 37	c_switch	perform C switch statement
IV - 38	c_uldiv	divide unsigned long by long
IV - 39	c_ulmod	remainder unsigned long by long
IV - 40	c_ulrsh	shift unsigned long right by count
IV - 41	c_ultd	convert unsigned long to double
IV - 42	c_unpk	unpack a double number