

INTERACTIVE

A Kodak Company

VP/ix Technical Guide

Software License Agreement

First printing (August 1988) PF073-00-0

No part of this manual may be reproduced in any form or by any means without written permission of:

INTERACTIVE Systems Corporation
2401 Colorado Avenue, 3rd Floor
Santa Monica, California 90404

© Copyright INTERACTIVE Systems Corporation 1985, 1988

© Copyright Phoenix Technologies Ltd. and INTERACTIVE Systems Corporation 1984, 1985, 1986, 1987, 1988

RESTRICTED RIGHTS:

For non-U.S. Government use:

These programs are supplied under a license. They may be used, disclosed, and/or copied only as permitted under such license agreement. Any copy must contain the above copyright notice and this restricted rights notice. Use, copying, and/or disclosure of the programs is strictly prohibited unless otherwise provided in the license agreement.

For U.S. Government use:

Use, duplication or disclosure by the Government is subject to restrictions as set forth in subdivision (b)(3)(ii) of the Rights in Technical Data and Computer Software clause at 52.227-7013.

All rights reserved. Printed in the U.S.A.

386/ix is a trademark of INTERACTIVE Systems Corporation.

VP/ix is a trademark of INTERACTIVE Systems Corporation and Phoenix Technologies Ltd.

UNIX is a registered trademark of AT&T.

Intel is a registered trademark of Intel Corporation.

80386 and MULTIBUS are trademarks of Intel Corporation.

AT and IBM are registered trademarks of International Business Machines Corporation.

Lotus is a registered trademark of Lotus Development Corporation.

Microsoft and MS-DOS are registered trademarks of Microsoft Corporation.

Programs described in this manual are copyrighted and their copyright notices may be found in heralds, by using the UNIX *what* program, and by reading files whose names start with "coprisc".

VP/ix Technical Guide

CONTENTS

1. INTRODUCTION	1
1.1 Functional Description of the VP/ix Environment	1
1.2 Components of the VP/ix Environment	2
1.2.1 VP/ix Emulation Control Process Support	2
1.2.2 Access to Devices	2
1.2.3 Channels	2
1.2.4 User Interface	2
1.2.5 Cross-Execution Facilities	2
2. THE VP/ix EMULATION CONTROL TASK	3
2.1 Kernel Interface Definition	3
2.2 Definition of a Dual-Mode Process	3
2.2.1 Creation of a Dual-Mode Process	4
2.3 Mode Transitions and the IDT	6
2.3.1 Kernel Trap Handling	6
2.3.2 Task Switching	7
2.4 The Pseudo-Interrupt Mechanism	8
2.5 The XTSS	9
2.6 Fork and Exec	11
3. ACCESS TO DEVICES	11
3.1 Emulation of Standard Devices	11
3.1.1 Keyboard Emulation	12
3.1.2 Display Emulation	12
3.1.3 Floppy Emulation	13
3.1.4 COM Emulation	14
3.2 Direct Attachment of Nonstandard Devices	15
3.3 Emulation Of Nonstandard Devices	16
3.3.1 Cooperation Between Driver and ECT Emulation Module	16
3.3.2 VP/ix Emulation Modules and the Generic Device Interface	17
3.4 Packaging and Distribution of Emulation Modules	27
3.4.1 VP/ix Configuration Tools	27
3.4.2 Contents of an IEM package	27
3.4.3 Creating the IEM	27
3.4.4 Packaging the IEM for User Installation	28
4. CHANNELS	28
4.1 Components of the Channels Package	29
4.2 Using the Channels Package	29
4.3 DOS Channel Functions	29
4.4 UNIX Channel Functions	31
4.5 A Typical Communication Session	32
4.6 Sample Application	33
5. USER INTERFACE	33
6. CROSS EXECUTION	33
6.1 Executing DOS programs from UNIX	33
6.2 Executing UNIX programs from DOS	34
Appendix A: DOS Modifications	35
Appendix B: INCLUDE FILE v86.h	37
Appendix C: ADAPTING A SERIAL TERMINAL DRIVER FOR USE WITH VP/ix	41
1. INTRODUCTION	41

2. GENERAL COMMENTS	41
3. KEYSTROKE NOTIFICATION HOOKS	41
4. IOCTL SUPPORT REQUIRED FOR VP/ix	42
Appendix D: GLOSSARY OF TERMS	45
Appendix E: SYSTEM CALL MANUAL PAGES	47
1. INTRODUCTION	47
Appendix F: INSTALLABLE EMULATION MODULES TOOLS MANUAL PAGES	53
1. INTRODUCTION	53

VP/ix Technical Guide

1. INTRODUCTION

This document provides a technical description of the VP/ix™ Environment which has been jointly developed by INTERACTIVE Systems Corporation and PHOENIX Technologies Ltd. It is intended for use by OEM customers who wish to understand the internals of the VP/ix Environment, or to customize it. For a higher-level, less technical description of the VP/ix Environment, see the *VP/ix Product Brief*. It is assumed throughout this document that the reader is familiar with the architecture of the Intel® 80386™ microprocessor.

1.1 Functional Description of the VP/ix Environment

The VP/ix Environment provides users of Intel 80386-based computer systems with the ability to run most MS-DOS® (referred to in this guide as DOS) applications under UNIX®. Initial versions of the product operate under UNIX System V Release 3 on 80386 hardware environments that are otherwise compatible with the IBM® PC AT®. However, the VP/ix Environment is expressly designed to be portable to *any* single-user or multiuser 80386-based UNIX system. A MULTIBUS™ version of the product is also available for customization by OEM's. That version is fully functional except for certain hardware-dependent features, such as diskette copy protection support, that are likely to change from system to system. Graphics applications require the appropriate display hardware, and are currently supported only on the PC AT version of the product. However, text-only applications will run under the VP/ix Environment on serial terminals as well as on memory-mapped video displays. Any serial terminal with 24 or 25 visible lines is supported.

Multiple DOS applications, including multiple instances of the same application, can run concurrently with conventional UNIX processes under the VP/ix Environment. The UNIX kernel provides the same efficient interrupt-driven scheduling services to DOS applications and UNIX applications alike, making the VP/ix Environment a true multitasking system.

The VP/ix combination of DOS and UNIX System V.3 provides the following features:

- Support for all facilities and applications normally available under UNIX.
- Support for all facilities and applications normally available under DOS, including support for the unusual hardware accesses required by *ill-behaved* programs.
- Transparent integration of the DOS and UNIX file systems. Any program, regardless of whether it is a DOS or a UNIX program, can access (and share) any DOS or UNIX file without concern for the boundaries between file systems or disk partitions. No partitioning is required.
- The ability for each user to run DOS and UNIX sessions simultaneously in a paged virtual memory environment. Each session runs in its own secure virtual address space.
- The ability to run DOS programs from a UNIX session and UNIX programs from a DOS session.
- Access to DOS applications from any serial terminal, not just the system console.
- Virtual screen support on the PC console permits multiple DOS and UNIX applications for the same user to run as though each one were in full control of its own screen. Up to eight applications running on virtual screens may share a physical screen. The user switches the physical screen from one virtual screen to another with a keystroke. Virtual screens are not supported on serial terminals.
- Built-in emulation support for standard IBM PC AT peripherals, including graphics adapters, mouse, modems, printers, etc.
- Simulated direct access to IBM-compatible peripherals for ill-behaved DOS programs on non-compatible systems, via a well-defined generic driver interface that facilitates the porting of the VP/ix Environment to new systems.

- Optional provision of 2MB of virtual EMS (Lotus®-Intel-Microsoft®'s *Expanded Memory Standard*) to any DOS application, regardless of a system's actual physical memory configuration.

1.2 Components of the VP/ix Environment

The VP/ix Environment consists of several functional areas. These areas are outlined briefly here, and discussed in detail in the remainder of this document.

1.2.1 VP/ix Emulation Control Process Support. The VP/ix Environment allows DOS and its applications to run under UNIX by creating a virtual machine in which all the standard layers of a real PC, i.e., the 8086 microprocessor, the surrounding hardware environment, and the BIOS, are available in either real or emulated form. A VP/ix virtual machine consists of a user-level program called *vpx* and some supporting modifications in the UNIX kernel. The bulk of the emulation in the VP/ix Environment is done in *vpx* rather than in the kernel.

vpx runs as a special kind of UNIX process called a *dual-mode* process. A dual-mode process spends part of its time in Virtual 8086 mode running DOS and its applications, and the rest of its time in Protected mode acting as the virtual machine monitor for its "other self." Each half of the process is a separate task in the 80386 hardware sense, in that each has a separate TSS. Virtual machines and dual-mode processes are one-to-one.

The technical details of dual-mode process support are discussed in section 2.

1.2.2 Access to Devices. A complete virtual PC must provide virtualized hardware support for the standard IBM PC environment. This includes such things as the keyboard, displays, serial lines, and parallel ports, but if ill-behaved programs are to be supported, it must also include timers, interrupt controllers, diskette controllers, and DMA controllers. The VP/ix Environment emulates all standard PC hardware, and runs all popular DOS applications, ill-behaved or not.

Device emulation is shared between *vpx* and kernel drivers.

There is also a class of devices to which it is safe to allow DOS drivers direct access, without any intervening emulation or UNIX driver. The VP/ix Environment provides a means to allow such access.

The VP/ix Environment also provides a way to add emulation for more complex non-standard devices such as high-resolution displays and network cards. Adding device emulations to VP/ix closely parallels (and sometimes requires) adding device drivers to the UNIX kernel; thus it is a task for the OEM or other hardware manufacturer, rather than the end user.

The details of device access, including how to package a set of drivers for a non-standard device for user installation and customization of the VP/ix Environment, are discussed in section 3.

1.2.3 Channels. The VP/ix Environment provides a mechanism whereby DOS applications which "know" they are running under VP/ix may communicate with the virtual machine monitor, and through it, with other UNIX processes. Thus an OEM may add communications code to a DOS application they are writing, and similar code in a pseudo-device emulation module they add to the virtual machine monitor, and use this communications channel to pass information back and forth to other UNIX processes using normal UNIX inter-process communications facilities. Or the channel may be used for the DOS application to request services from the virtual machine monitor.

The specification of this mechanism is given in section 4.

1.2.4 User Interface. The VP/ix pop-up User Interface Menu allows users to communicate directly with *vpx*. This is discussed in section 5.

1.2.5 Cross-Execution Facilities. The VP/ix Environment permits users to execute DOS programs directly from the Bourne or C shells, and UNIX commands from the DOS shell (COMMAND.COM). These facilities are discussed in section 6.