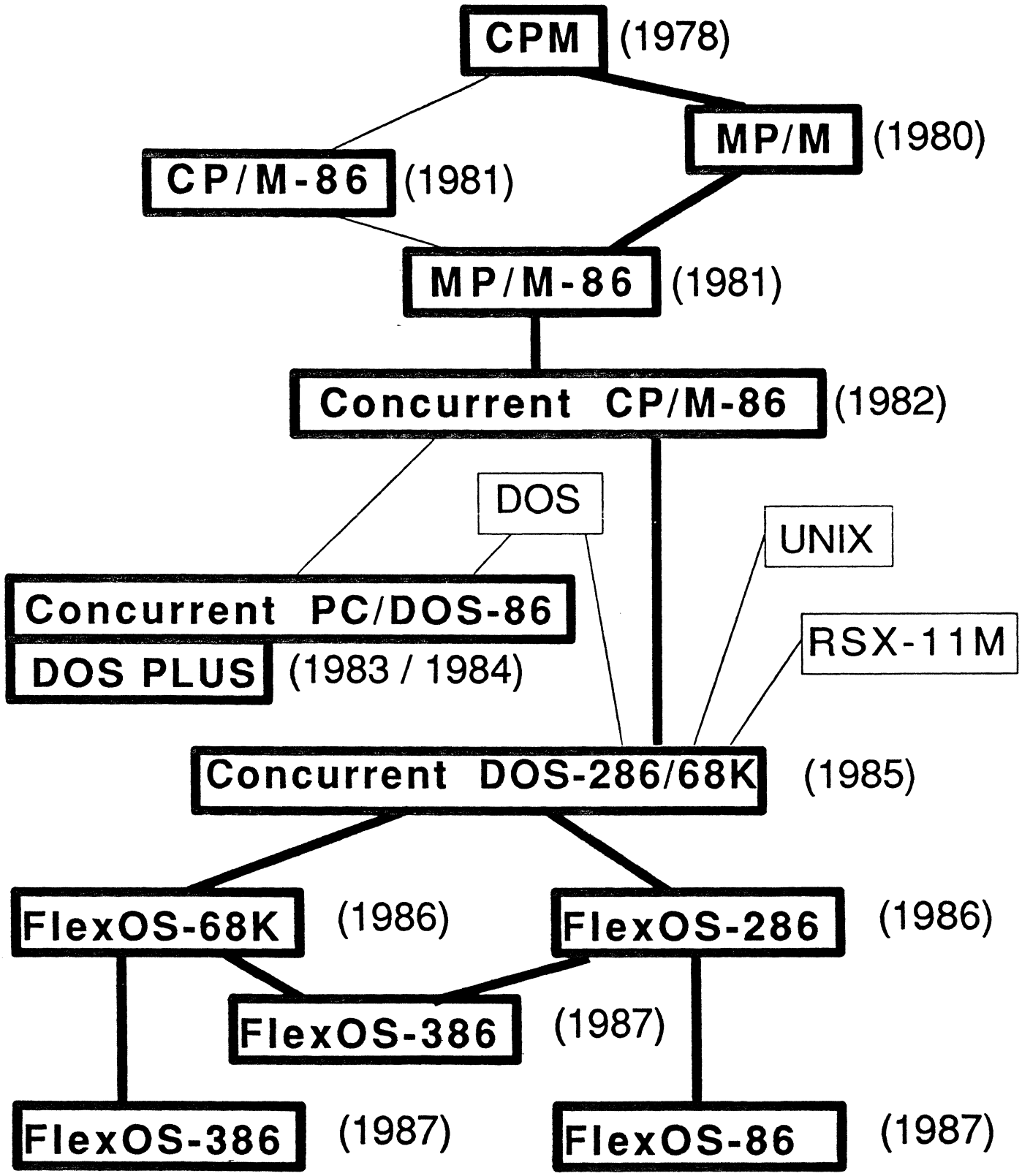
FlexOS

The Flexible Automation Operating System

TECHNICAL PRESENTATION

Contents

Development History



Features

Process Management

- Real-Time
- Multi-Tasking
- Single or Multi-User

Protection

- Protected File Access
- Protected Memory

Async I/O System

- 31 Events per process
- Wait for multiple events
- Software Interrupts on event completion

Configuration

- Dynamically Loadable Device Drivers
- Modular

Console System

- Virtual Consoles
- 16-bit/8-bit Character Sets
- Standard Terminal I/F

Graphics

- Standard VDI Interface

Disk System

- Shared File System
- Hierarchical Directories
- DOS 3.x Compatible
- File ownership

Networking

- IBM PC Networking Compatible (SMB)
- Transparent Access

Portability

- CPU and Device Independent Interface

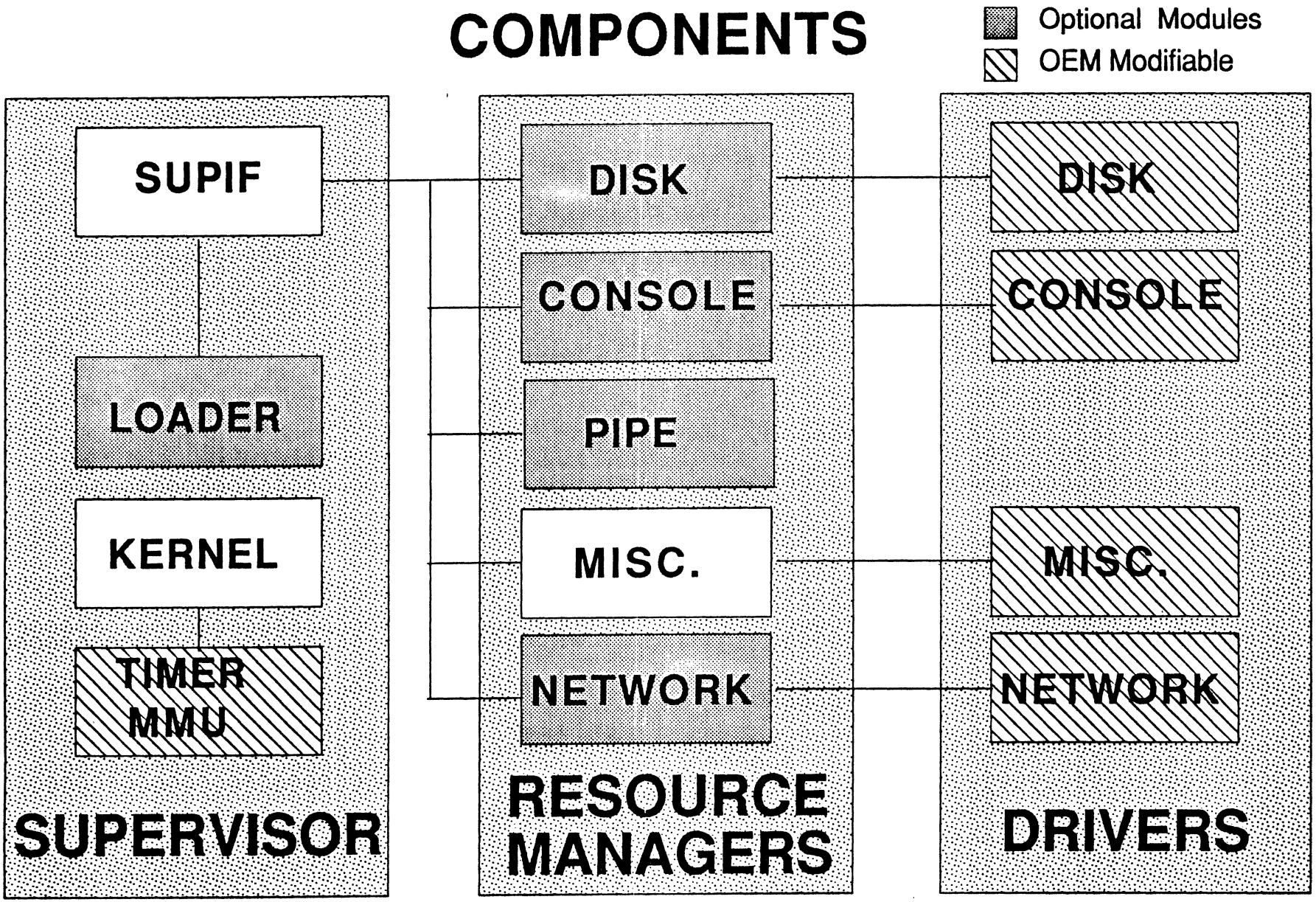
Front Ends

- PC DOS 2.1
- CPM-68K

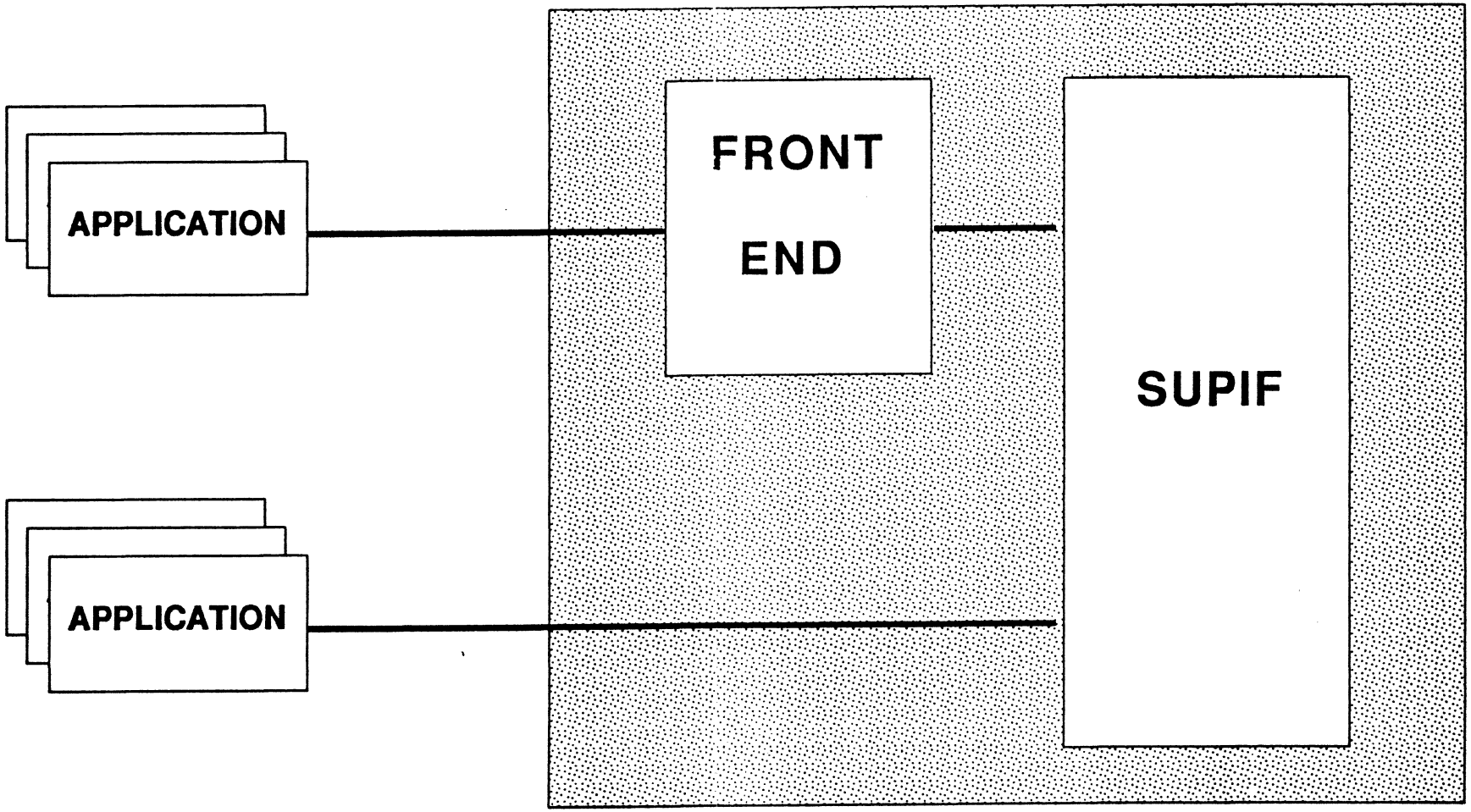
International Considerations

- International Character Sets
- Customizable Messages

COMPONENTS


Optional Modules
OEM Modifiable
SUPIF
LOADER
KERNEL
TIMER
MMU
SUPERVISOR
DISK
CONSOLE
PIPE
MISC.
NETWORK
RESOURCE
MANAGERS
DISK
CONSOLE
MISC.
NETWORK
DRIVERS

COMPONENTS



The File System (Supervisor)

Controls Access to all Resources

Driver Installation and Removal

All Files and Devices are Named

Logical Name Substitution

Concurrent File Access Protection

Protected Access by User/Group ID

Unique Table for each File Type

File Naming

[node::]device:[\][directory\]...[filename]

node::	network node name
device:	logical device name
\	root directory
directory\	subdirectory name
filename	file name (and extension)

Logical Names

- DEFINE SVC
- Prefix Substitution
- Literal option
- Up to 99 levels of definition
- default: applied to filenames without node and device names
- system: applied if default fails, must have SYSTEM attribute

Examples:

home:	hd0:\mydir\
con:	con2:\vc003\
con	con:console
prn:	prt2:
stdin	con
stdout	prn:
stderr	con
path	default: home: system:
prompt	tg

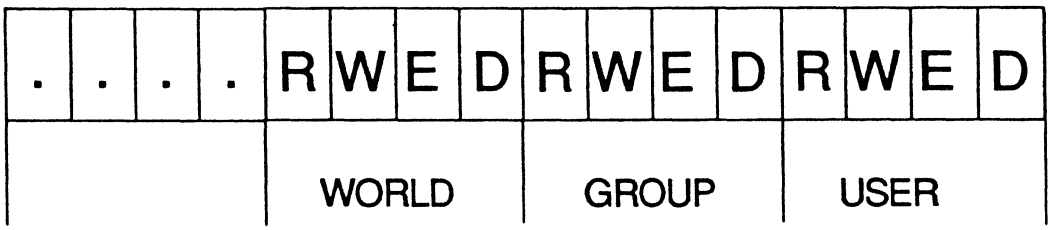
Kinds of Files

DISK	Disk Device Control, RAW I/O
DISKFILE	Disk File I/O, Control
PCONSOLE	Physical Console Control
VCONSOLE	Virtual Console Control
CONSOLE	Keyboard and Screen I/O
LEFT, RIGHT, BOTTOM, TOP	Window Border I/O
PIPE	Pipe I/O
SERIAL	Serial Device I/O and Control
PRINTER	Printer Device I/O and Control
PORT	Port Sub-Driver Control
SPECIAL DEVICES	Special Device I/O and Control
NETWORK DEVICES	Network Device Control

File Security

User LOGON determines User/Group ID

Each file is created with a SECURITY WORD:



Each OPEN attempt is validated by comparing the OPEN Access Request

with the SECURITY WORD and Previous Successful Opens

OPEN Access Requests:

Read	Shared/Exclusive
Write	Shared/Unique File Pointer
Execute	Reduced Access
Delete (Set)	

File I/O

OPEN, CREATE returns File Number

READ, WRITE, SPECIAL, CLOSE, GET and SET operates with File Number

I/O Control Block is Independent of type of file

Byte Level I/O

I/O Relative from:

- Beginning of File
- End of FIle
- File Pointer

Shared or Unique File Pointer

Record Locking

STDIN, STDERR, STDOUT

SVC Interface

__osif(function,parameter);

function = SVC function #
parameter = 32 bit number or
Parameter Block Address

286 Assembler:

CX = function

BX:AX = parameter

INT 220

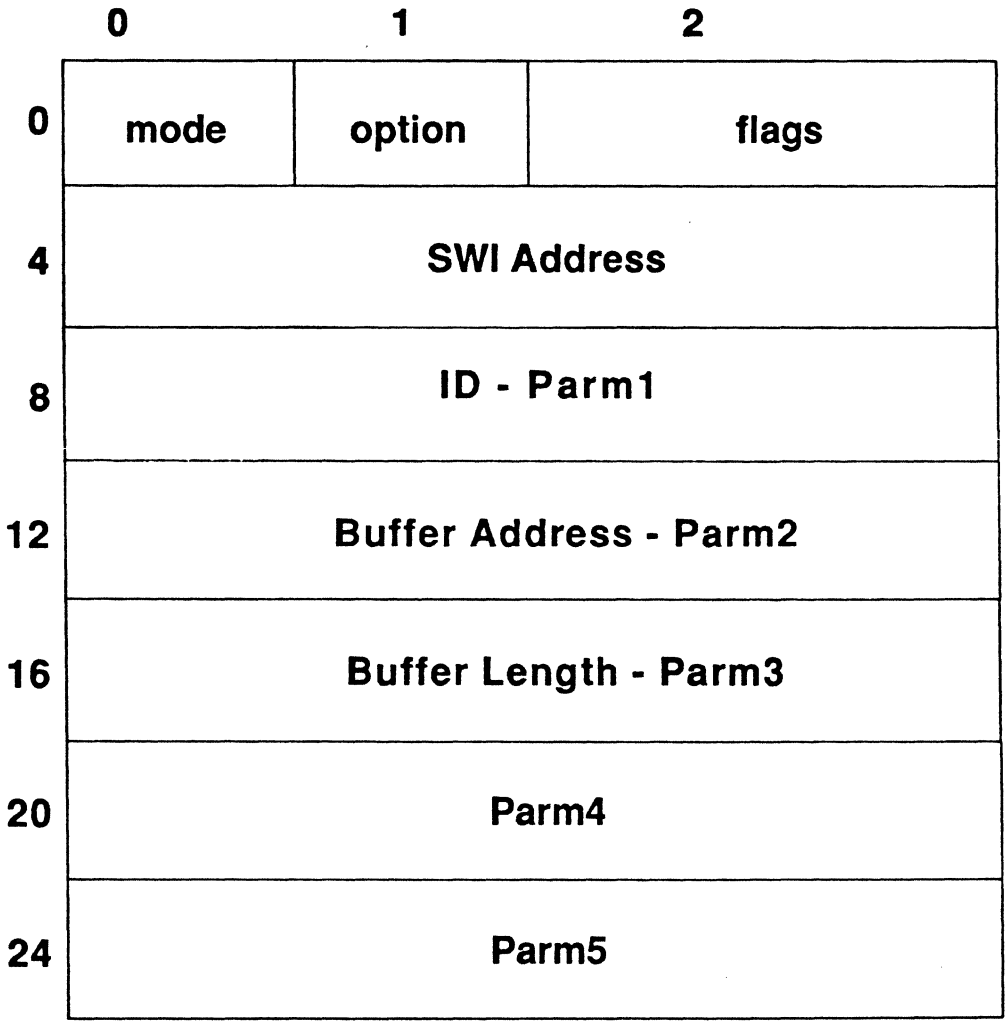
68K Assembler:

D0.W = function

D1.L = parameter

TRAP 14

SVC Parameter Block Format



File Management SVC's

DEFINE	Define Logical Name
CREATE	Create a File
DELETE	Delete a File
OPEN	Open a File
CLOSE	Close a File
READ	Read a File
WRITE	Write a File
SEEK	Modify/Obtain File Pointer
LOCK	Lock/Unlock Area of Disk File
RENAME	Rename or Move a File
COPY	Copy Screen Rectangle
ALTER	Alter Screen Rectangle

Device Management SVC's

SPECIAL	Perform Special Device Function
DEVLOCK	Lock/Unlock Device
INSTALL	Install/Replace/Link Drivers

Event Management SVC's

CANCEL	Cancel Events
WAIT	Wait for Events
STATUS	Get Event Status
RETURN	Get Event Completion Code

Process Events (Event Bits):

Created only in Process Context

Up to 31 Outstanding Events per Process

Each event represented by an Event Bit

WAIT - Wait for Multiple Events

RETURN - Obtain Return Code of Single Asynchronous Event

STATUS - Obtain Completion Status of outstanding Events (event mask)

CANCEL - cancel events (Event Mask)

Process Management SVC's

TIMER	Create Timer Event
ABORT	Abort Specified Process
COMMAND	Perform Program Load
EXCEPTION	Set Exception Trap
MALLOC	Allocate Memory
MFREE	Free Memory
EXIT	Terminate with Error Code
ENABLE	Enable SWI's
DISABLE	Disable SWI's
SWIRET	Return from SWI
CONTROL	Control a Process
OVERLAY	Load Overlay

Console Management SVC's

KCTRL	Obtain keyboard
ORDER	Order Windows
XLAT	Keyboard Translation
GIVE	Give Keyboard to Child

Table Management SVC's

GET	Get Table Info
SET	Set Table Info
LOOKUP	Wildcard Lookup of Tables

Resource Manager Tables

PIPE	Pipe File Information
DISK	Disk Device Information
DISKFILE	Disk file Information
PCONSOLE	Console Device Information
VCONSOLE	Virtual Console Information
CONSOLE	Console File Information
PRINTER	Printer Device Information
PORT	Port Device Information
SPECIAL	Special Device Information

Supervisor Tables

PROCESS	Process Information
ENVIRON	Process Environment Information
TIMEDATE	System Time and Date
MEMORY	System Memory Information
SYSTEM	Global System Information
FILNUM	Open File Information
SYSDEF	System level defined names
PROCDEF	Process level defined names
CMDENV	Command Environment
DEVICE	Global Device Information
PATHNAME	Expanded Path given logical name

System Configuration

CONFIG.C - System Build Options

- Resource Managers
- Static Drivers (minimum driver set)
- OS data pools
- Maximum Memory
- CONFIG Process Startup (Dynamic Memory Sizing etc.
- Initialization

BOOT:CONFIG.BAT - Boot Options

- Logon Protection Option
- Loadable Drivers
- Default Logical Names
- "Normal Shell Script"

SYSTEM:USER.TAB - LOGON Options

- User Names
- Default Window Manager and Shell
- Home Directory
- User/Group ID, Password

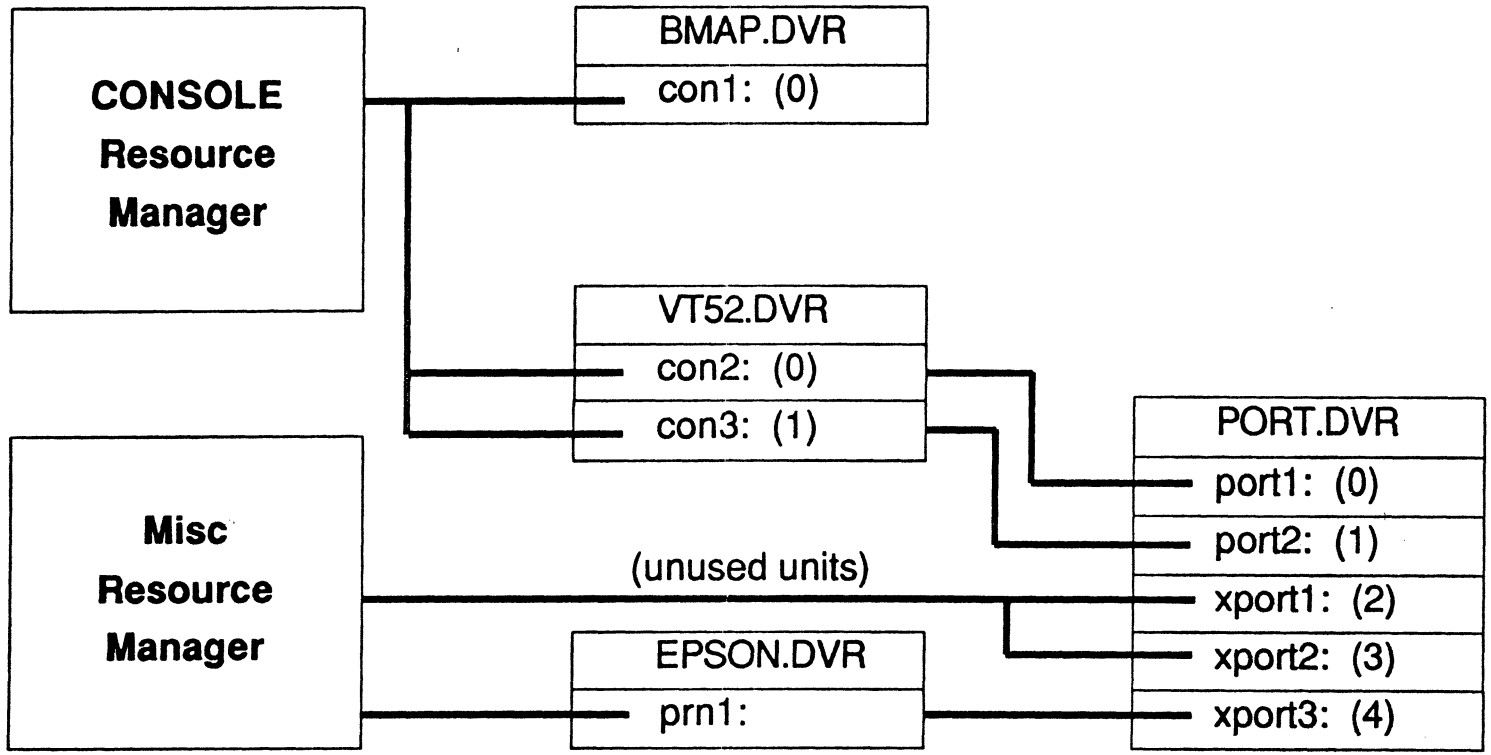
HOME:AUTOEXEC.BAT - Shell Startup Script

Device Drivers

- Static or Dynamic
- Dynamic Drivers can be Installed or Removed at any time
- Supports One or More Units
- Each Unit is a logical device
- Driver Units optionally synchronized at Driver or Unit level
- Driver Units are controlled by Resource Managers or

  by other Driver Units (becoming a Sub-Driver)

Example of Drivers and Sub-Drivers



Disk Resource Manager

PC DOS 3.x Media Compatible

Hierarchical Directories

Asynchronous Record Locking

File Ownership by User/Group ID

Directory Label

Fixed Length Records

Mixed Case Media

Console Resource Manager

Multiple Physical Consoles

Virtual Console Management

- Tree Structured
- Dynamically created and deleted
- Windows - Control Size, Placement, and View of Virtual Consoles

Standard Keyboard

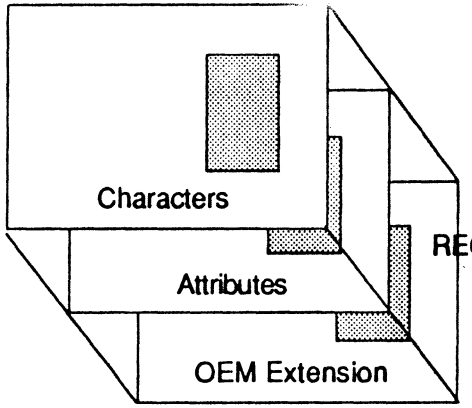
- VT52 plus Extensions
- Optionally Modified by Driver

Standard Screen

8 or 16-bit Modes for Input and/or Output

Supported Character Sets:

- US - 8-bit IBM PC
- Europe - 8-bit ISO ASCII
- Japan - 8-bit KANA
  - Shift-JIS KANJI
  - 16-bit KANJI



COPY/ALTER Screen Model

VFRAME - Driver Representation of Virtual Console

PFRAME - Physical Representation

Pipe Resource Manager

"pi:" device, named pipes

Interprocess Communications (Message Pipes)

Interprocess Synchronization (Semophore Pipes)

In memory buffers Only

Dynamically Created and Deleted

Security

Non-Destructive Reads

Temporary Pipe (Delete on Last Close)

Provision for "server" processes to WAIT

instead of EOF on CLOSE of opposite end

Network Resource Manager

Loadable Resource Manager and Drivers

Based on OSI/ISO Model

PC/NET 1.0 Compatible

Default and Extended security (LOGON)

Transparent Access to Remote Resources

OEM Transporter - Easy for an OEM to Port to their Hardware

Logical Node and Socket Naming

Graphics

Industry Standard VDI (Virtual Device Interface)
Compatible with GEM VDI
Device Types :
- Console
- Printer
- Plotter
- Camera
- Mouse
- Metafiles

Virtual Consoles
Windows / Borders
VDI drivers can be modified by OEM
to support diverse hardware
Example implementations include:
- IBM PC/AT CGA
- IBM PC/AT EGA (10 colors)
- Hercules Mono Card
- EPSON High / Low Res Graphics Printer
- Mouse Systems Mouse
- Summamouse Mouse

VDI Functionality

- GSX SVC
- Color or Mono
- Normalized Device Coordinates
  - Device Independent Operations
  - Automatic Transformation of NDC to RC
- Raster Coordinates
  - High Performance
- Alpha and Graphic Modes
- Vextor Operations
  - Line
  - Circle
  - Elipse
  - Pies
  - Rectangles
  - Arc
  - Pattern Fill
- Raster Operations
  - BIT-BLT
- Multiple Character Fonts

OS Memory Model

User Memory Space

(running user code)

CODE	Loaded Non-modifiable Sharable
DATA	Loaded Non-executable
STACK	Created at Load Non-executable
HEAP	Dynamic Non-executable

System Memory Space

OS code and data

Driver Code and Data

SALLOC'd memory

currently mapped user space

(running system code)

Mapped and Protected MMU's

Dynamic First Fit Allocation of TPA and logical Address Spaces

User Space, System Space and Physical Memory are Independent Address Spaces

Driver Services provide:

- Address Space Conversion
- User Space Remapping
- User Space Locking
- Addressability to Physical Memory that is not in TPA

Interrupt Service Routines (ISR)

Global Interrupt Stack (4K)

SETVEC initializes Interrupt Vectors

Nested ISR's Allowed

DOASR - Only OS Service allowed

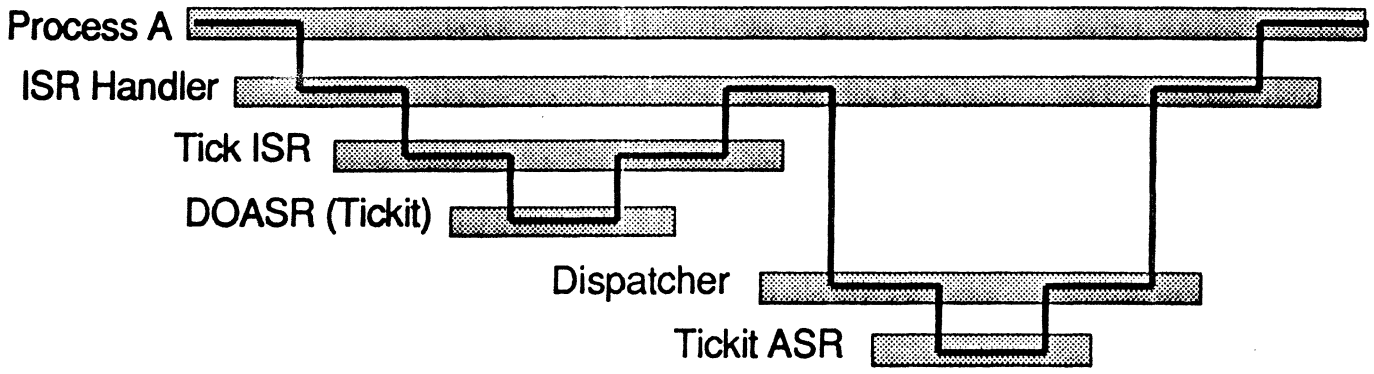
Machine State automatically Saved/Restored

ISR can be written in 'C'

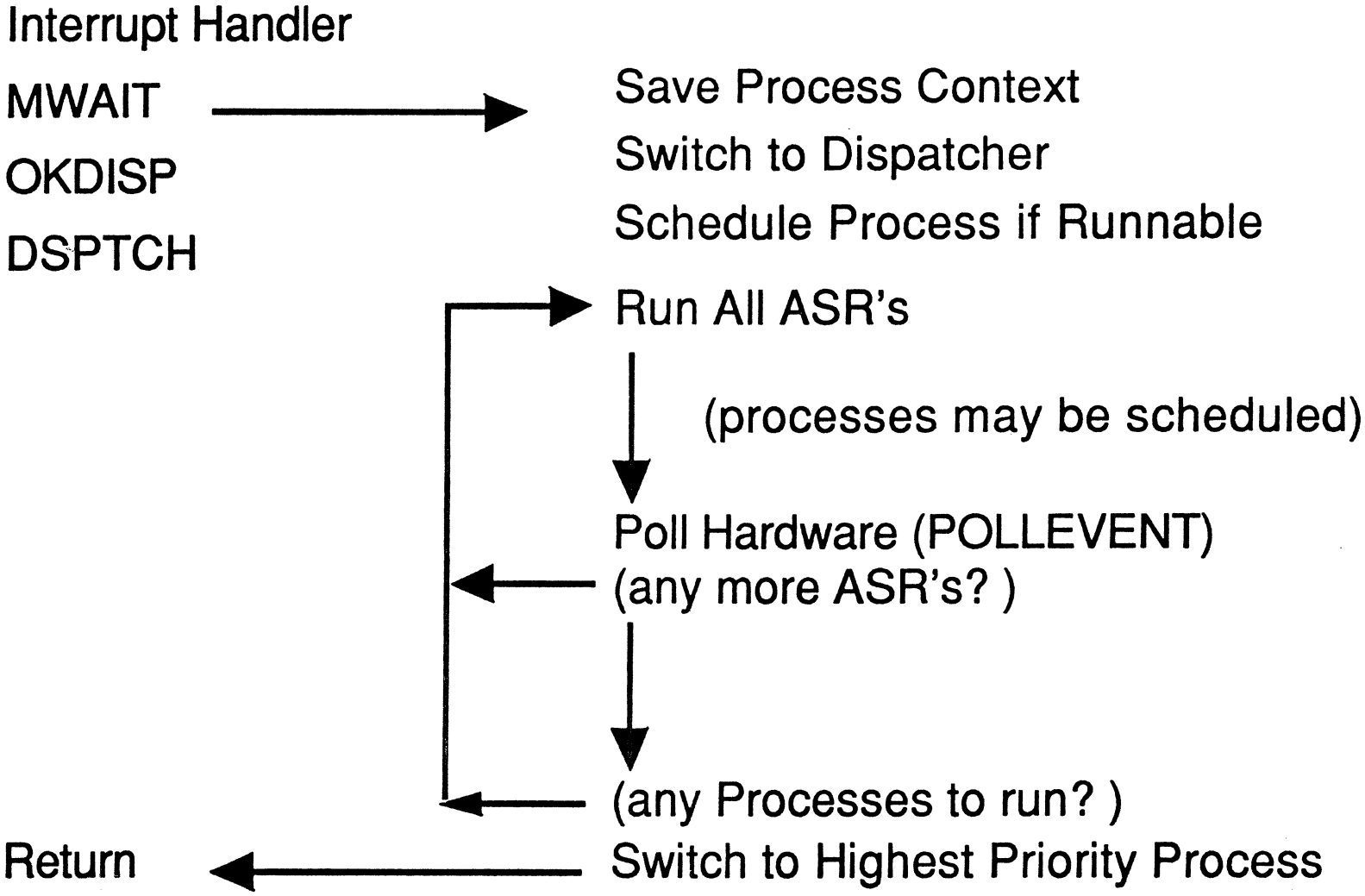
"Force Dispatch" Option

Philosophy: Keep ISR short, use ASR for Extra Work

Tick ISR Overhead when completing Timer Event



Dispatcher


Interrupt Handler
MWAIT
OKDISP
DSPTCH
Save Process Context
Switch to Dispatcher
Schedule Process if Runnable
Run All ASR's
(processes may be scheduled)
Poll Hardware (POLLEVENT)
(any more ASR's?)
(any Processes to run?)
Switch to Highest Priority Process
Return

Asynchronous Service Routines (ASR)

- Second Level Interrupt System
- Runs under Dispatcher Context
- Prioritized Scheduling, FIFO if equal
- ASR's always runs to completion (except ISR's)
- Scheduled from ISR, ASR or process
- All scheduled ASR's run before processes
- ASR's can be scheduled to run upon Event Completion
- ASR's cannot WAIT

ASR Events (Event Numbers):

Created in ASR context

Unlimited # ASR Events

NEXTASR - Schedule ASR on Event Completion

RETURN - Obtain Completion Status (Event must be complete)

STATUS - Obtain Status of single Event

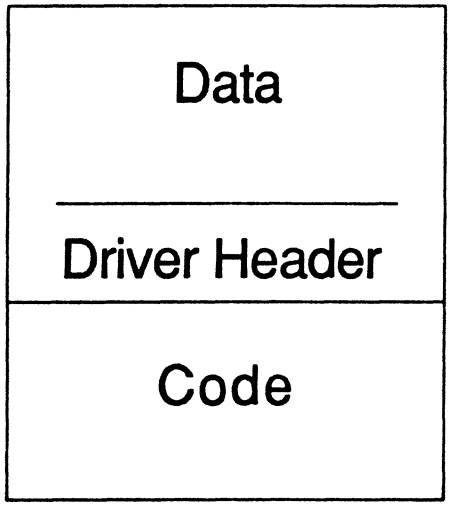
Cancel - Cancel single ASR Event

EVASR - Convert Process Event to ASR Event and Schedule ASR

Kernel Timings

	Measured on IBM PC/AT at 6 MHz with 1 memory wait state	Estimated time at 8 MHz with 0 wait states (.6 x measured time)
Process Dispatch (Different Process)	.309 msec	.185 msec
Process Dispatch (Same Process)	.247	.148
Tick Overhead	.474 + Tick ISR	.284 + Tick ISR
ISR Handler Overhead (no dispatch)	.080	.048
DOASR	.070	.042
Interrupt to ISR	.060	.036
ISR to ASR	.163	.098
Last ASR to Process (Diff. Process)	.200	.120
Last ASR to Process (Same Process)	.138	.083
FLAGEVENT	.359	.215
FLAGSET	.242	.145

Driver Header



Driver Header Contains:

Maximum Number of Units Allowed

Synchronization Level Needed

INIT Address

SUBDRIVE Address

UNINIT Address

SELECT Address

FLUSH Address

READ Address

WRITE Address

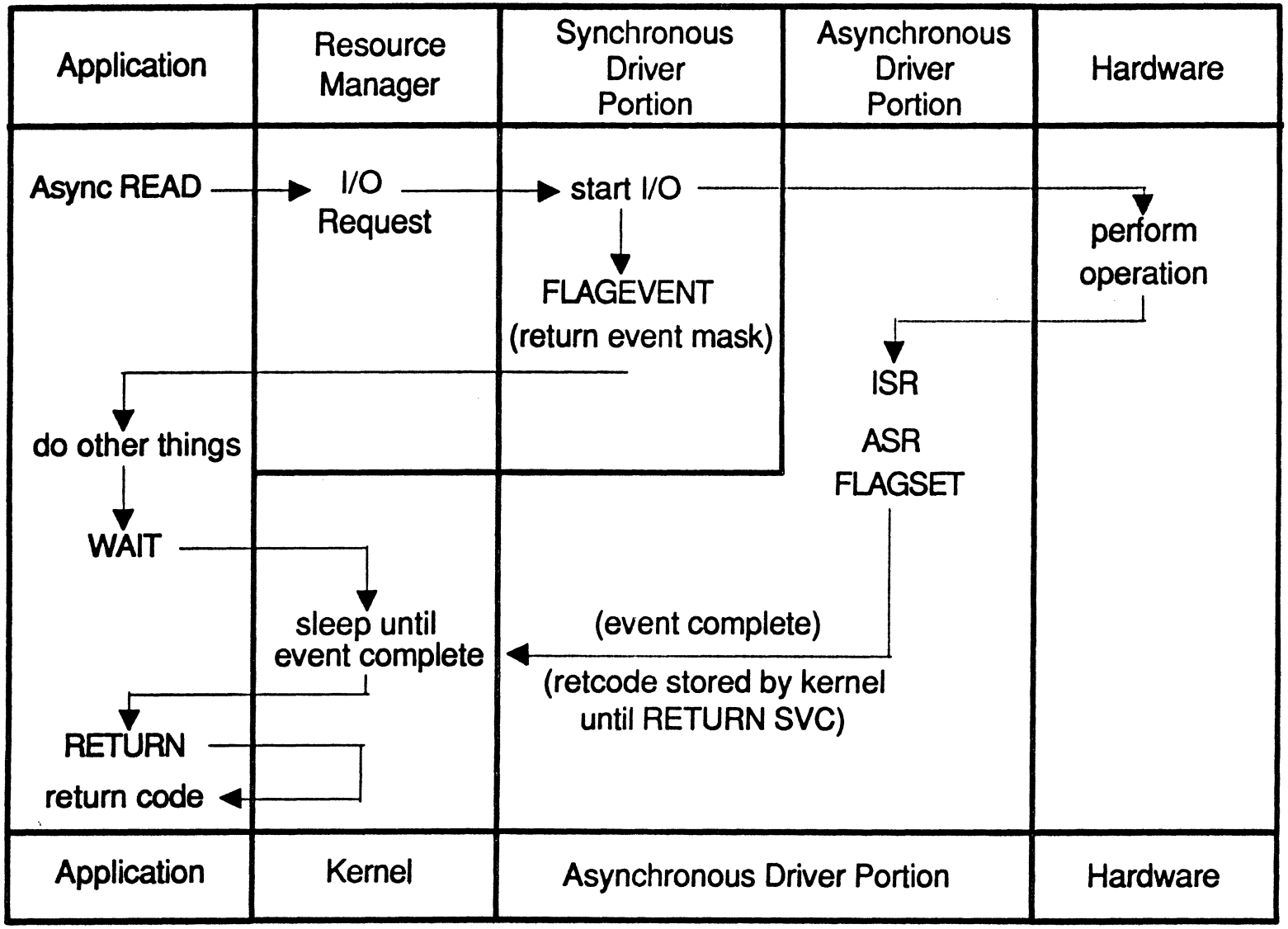
GET Address

SET Address

SPECIAL Address

Pointers to System Variables

Asynchronous Driver I/O



Driver Services

Flag System

FLAGCLR	Clear a System Flag
FLAGEVENT	Get Flag Event
FLAGGET	Allocate System Flag
FLAGREL	Free System Flag
FLAGSET	Satisfy a Flag Event

Device Polling

POLLEVENT	Get Hardware Pool Event

Memory Management

MAPU	Map User Memory
MAPPHYS	Map Physical Memory
MLOCK	Lock User Memory
MRANGE	Check Buffer Range
MUNLOCK	Unlock User Memory
PADDR	Convert System to Physical
SADDR	Convert User to System
SALLOC	Allocate System Memory
SFREE	Free System Memory
UADDR	Convert System to User
UNMAPU	Re-map User Memory

Driver Services

ASR Management

ASRWAIT	Wait for ASR Event
DOASR	Schedule an ASR
DSPTCH	Force a Dispatch
EVASR	Schedule ASR on Proc. Event
NEXTASR	Schedule ASR on ASR Event

Critical Regions

ASRMX	Obtain MX Region
MXEVENT	Obtain MX Region
MXINIT	Create MX Region
MXREL	Release MX Region
MXUNINIT	Delete MX Region
NOABORT	Enter No Abort Region
NODISP	Enter No Dispatch Region
OKABORT	Exit No Abort Region
OKDISP	Exit No Dispatch Region

Other

PCREATE	Create Process
SETVEC	Initialize Interrupt Vector
SUPIF	Make Supervisor Call

Utilities and Built-in Commands

ASSIGN	Assign Logical Drive Name
BACK	Start Background Command
BACKUP	File Backup
BANNER	Print Banner
CANCEL	Terminate Process
CAT	Copy STDIN to STDOUT
CHDIR	Change Default Directory
CHKDSK	Check Disk Integrity
COMMAND	User Shell
COMP	Compare files
CONFIG	Configure Serial Port
COPY	Copy Files
CTTY	Change STDIN and STDOUT
CUT	Cut Fields from File
DATE	Display/Set Date
DEFINE	Define Logical Name
DIFF	Display File Differences
DIR	Directory
DISKCOMP	Compare Disk Devices
DISKCOPY	Copy Disk Image
DISKSET	Set Disk Device Information
DUMP	Display File Contents
DUP	Print String Multiple Times w/Variables
ECHO	Echo command Tail
ERASE	Erase Disk File
EXIT	Exit Shell
FDISK	Prepare Hard Disk
FGREP	Find Exact Strings
FIND	Find string Pattern
FORMAT	Format Disk
FSET	Set Disk File Attributes
GREP	Find Strings matching Regular Expression

Utilities and Built-in Commands

HSET	Set Shared Code in .286 File
LC2UC	Lower Case to Upper Case
LIST	List Built-in Commands
LOGON	Login as a different User
LOGOFF	End Logon Session
MKDIR	Create a Disk Directory
MORE	Display File One Screen at a time
ORDER	Set Command Type Search Order
PASSWORD	Change Login Password
PASTE	Merge Files Line by Line
PATH	Set Command Search Path
PR	Format STDIN to STDOUT
PROCESS	Display Current Tasks
PROMPT	Change shell Prompt
RECDIR	Repair Disk Directory
RECFILE	Repair Disk File
RENAME	Rename or Move Disk Files
RESTORE	Restore BACKUP Files
RMDIR	Remove Disk Directory
SECURITY	Set Default Security Word
SORT	Sort STDIN to STDOUT
SPLIT	Split a File
STRINGS	Display Strings in File
SUM	Checksum a File
SYS	Build Boot Disk
TIME	Display/Set Time
TREE	Display Directory Paths
TYPE	Display File
UC2LC	Upper Case to Lower Case
VER	Display OS VersionString
VOL	Display Volume Label
WC	Count Words/Lines in File

Development Environment

Native Development

	286	68K
C Compiler	Metaware	Alcyon
Linker	LINK86	LINK68
Librarian	LIB86	AR68
Assembler	RASM86	AS68
Standalone Debugger	ATLD	SAS
Process Debugger	SID	SID