DOS NOTES

DELIQUENT OPERATING SYSTEM

ANNOTATION

Taking an observation of programming is important to understand the basics of the IBM operating system DOS AND MSDOS. In command prompt following the functions file and file extensions of delinquent operating system. Command prompt can be found in search at the bottom of the drop down menu for applications. Type in “command prompt” here is some helpful commands and hierarchy about DOS. Annotated data summarizing the program language used by internal business machine(IBM).

- NOVELL NETWARE ARE BUILT ON DOS
- DOS CORE OF DEVELOPMENT

  WINDOWS
- MS-DOS AND RXDOS- UNDERSTANDING OF THE MECHANICS, ALGORITHMS,DESIGN AND STRUCTURE OF MS-DOS
- CONFIG OR AUTOEXEC SPECIAL LITTLE PROGRAM OF DRIVERS HERE AND THERE
- OTHER SYSTEMS (UNIX)(MACINTOSH)
- MSDOS.SYS AND IO.SYS , MAKE UP THE CORE OF DOS OPERATING SYSTEM
  - CONTAIN EVERYTHING THAT APPLICATIONS VIEW AS DOS
  - INCORPORATE- DEVICE DRIVERS, FAT MANAGEMENT (PARTITION DRIVE), DIRECTORY SEARCHING, DATE AND TIME SERVICES, CRITICAL ERROR HANDLING, DISK BUFFER CACHING, .INT FUNCTION HANDLER, MEMORY MANAGEMENT SERVICES AND PROGRAM LAUNCHING
  - COMBINED WITH COMMAND SHELL, COMMAND .COM THE ESSENTIAL FILES REQUIRED TO RUN DOS
- IBM VERSION OF DOS IOMDOS.COM AND IBMBIO.COM , REPLACES MSDOS.SYS AND IO.SYS RESPECTIVELY THE NATURE REMAINS THE SAME.

The core DOS files are usually stamped as hidden or system files, which would make them invisible to the casual user. The attributes can be changed for these files using the DOS – supplied attrib. program or a really good disk viewer like xtree of Norton’s diskedit.

COMPONENT FILE NAMES

MICROSOFTS MS-DOS	IBM’S PCDOS	RXDOS
IO.SYS	IBMBIO.COM	RXDOSBIO.COM
MSDOS.SYS	IBMDOS.COM	RXDOS.COM
COMMAND.COM	COMMAND.COM	RXDOSCMD.COM

The first module loaded at boot time is the io.sys or (rxdos.rxdosbio.com), which is responsible for interfacing dos to the hardware layer through residat device drivers and for system initialization and configuration.

- RESIDENT DEVICE DRIVERS
    - CONSOLE
    - PRINTER
    - COMMUNICATIONS
    - CLOCK
    - ONE BLOCK DEVICE CAPABLE OF ACCESSING THE FLOPPY DISK DRIVES AND STANDARD INTERFACE HARD DISK

First task performed by the io.sys code is configuring and initializing the dos environment and data areas. This is performed by the sys init, a replaceable component of io.sys that loads msdos.sys, initializing resident drivers, determining the amount of memory, processing the config.sys file and then loading the command shell. The init process may cause the loading of installable dos drivers, a concept introduced to dos with version 2.0.

- MSDOS.SYS – is always veiwed as being the bulk of dos because it supports the fat file system, all int services, device redirection, memory management, disk cache buffering, product launching and termination.
    - Rxdosbio.com – contains all drivers that are supplied standard with dos ; disk drivers, line printer, communication, keyboard and screen drivers.
    - Rxdos.asm – contains all of the init functions even thoughh some of them are just shells to functions handled in other modules.
    - Rxdosdev.asm – contains device interface code, such as locating devices in rxdosbio module
    - Rxdosexe.asm – launches .com and .exe programs
    - Rxdosfat.asm – provides all the logic fat management, such as allocating, searching, and fat entries
    - Rxdosfcb.asm – file control block support

- Rxdosfil.asm – file support, navigation, sub directory management, including searching, creating and removing subdirectories, and file services that search, open, create, read and write close, delete and position ()seek
- Rxdosint.asm – dos initialize
- Rxdosmem.asm – memory management, allocation, and free with composition
- Rxdossft.asm – system file table support; part of the file system supported
- Rxdosstr.asm – string management such as upper / lower case compare , and append
- rxdosstk.asm - stack managment

rxdoscmd.com - command

- rxdoscmd.asm – main body of command module, including command parsing, batch file starter and file redirection
- rxdoscpy.asm – copy command
- rxdosdir.ase – dir command
- rxdosfor.asm – supports batch file for command
- rxdosprm.asm – prompt command
- rxdosren.asm – rename command
- rxdoscsk.ams – command module stack

- .int 21 h function begin in the rxdos.asm source file – also contains the function dispatcher
- Rxdos functions are largely group together by subsystems
- Memory support functions appear in rxdos – mem.asm whereas the int 21 functions 49h, 4ah and 4bh (allocate, free and resize) start in rxdos.asm module , as do all the 21 functions. System file support functions appear in the rxdossft.asm module, but file function such as create and open begin in rxdos.asm

THE ROLE OF EACH DOS LAYER

COMMAND.COM

- COPY, RENAME, DEL
- MKDIR, CD, RD BATCH PROGRAMS

MSDOS.SYS

FAT FILESYSTEM	MEMORY MANAGMENT
FAT ALLOCATION	MEMORY ALLOCATION
SUB DIRECTORING	PROGRAM LOADING
REDIRECTION	TSR MANAGMENT

IO.SYS

LOGICAL SECTOR FILE SYSTEM

DEVICE DRIVERS

SYSTEM INITIALIZATION

BIOS (NOT PART OF DOS)

HEAD, SECTOR, TRACK DISK IO

The rxdos.com begins with a section data area, followed by the function handler table, followed in turn by each DOS function and related routines.

- Rxdosbios.com module – code is always loaded first by the startup code and begins at location 70:0
- Rxdos.com after which the stacked space is allocated
- Memory from the top of the stack space to the end of available memory becomes the memory allocated through dos

Dos swap area, this space is considered so critical that it is sometimes saved by being swapped out and then restored.

- Poor man's multitasking, where two processes can have different contexts and disk transfer access.
- Data areas begin before swap areas
- Data areas structures, dynamically allocate based on configuration parameters.
- System – initially populated by startup code allocating, and setting the system variable areas.
- Sysvar - contains pointers to the start of the device chain, to the system file tables to the allocatable memory chain, to drive parameter table – to other data structures.
- Access to the sysvar area and the Dos swap areas by an application are important because of the wealth of references.
- Sysvar changed from version 2.0 to 3.1 can change again – important list became a third party software developer.
- Quarterdeck , maker of QEMM and other memory – related utilities, wanted to relocate Dos buffers and file areas to unused upper memory blocks- that memory above 640 k – long before Dos version 5 made it possible to take advantage of upper memory blocks.

  Releasing of some critical memory in the lower 640k and enhanced performance be permitting greater buffers and files allocation as an undocumented list, a mess for the user.
- Application – programs can locate the DOS data segment address that contain the sysvar area – using Dos function 52h.

- Dos swap area is further up in the data segment and the adress to the start of the area and size is accessible through Dos function 5dh, subfunction 06h, address of Dos swappable area in a documented call.
-

-