

MS-DOS 7 Direction

Bens
6/10/93

Summary

This document describes two possible feature sets for MS-DOS 7 (*Chicago minus the GUI* versus *MS-DOS 6 + Features*), discusses the work items and disadvantages of each feature set, and then recommends that MS-DOS 6 + Features be our plan for MS-DOS 7.

Contents

1. Assumptions	1
2. Why should we produce MS-DOS 7?	1
2.1. Not all OEMs will preinstall Chicago.....	1
2.2. Take every last dollar of upgrade revenue.....	1
2.3. Competition from MS-DOS Clones.....	1
2.4. Asian acceptance of Windows is lagging U.S./Europe.....	2
3. Chicago minus the GUI.....	2
3.1. Features	2
3.2. Work Items.....	2
3.3. Disadvantages.....	3
4. MS-DOS 6 plus Features.....	3
4.1. Features	3
4.2. Work Items.....	4
4.3. Disadvantages.....	4
5. Recommendation	4

*vids Plug & Play dependent
PIP requires messaging user?*

"we are committed to synergy"

Bens

- new prot mode driver model
- no pnp
- no vid
- no mvdsm
- no multi-tasking
- no prot mode file system
- dblspace
- integrated network
- i.e. ms dos 6.5

bradi, brad c.

- common protmode driver model
- common " subsystem
- mvdsm
- Chicago w/ogoi

File: MS-DOS 7

Plaintiff's Exhibit

7681

Comes V. Microsoft

MS 0115609
CONFIDENTIAL

1. Assumptions

- o Chicago will ship before 9/94.
- o MS-DOS 7 will ship within 3 months of Chicago (either before or after)
- o We are very unlikely to release another major version of MS-DOS after MS-DOS 7
- o Chicago runs OK on a 4Mb 386 machine, but really wants an 6-8M 486 machine.

2. Why should we produce MS-DOS 7?

Most of Microsoft is focused on our strategic operating system platforms: Chicago, Windows NT, and Cairo. This is certainly justifiable, as the future lies with graphical user interfaces, 32-bit operating systems, and object-oriented technology. However, we have historically overestimated that rate of adoption of new operating systems by our customers. There are still a significant number of customers who will want and need MS-DOS (or a clone product from another company) even after Chicago is released.

2.1. Not all OEMs will preinstall Chicago

2.1.1. To Reduce Cost

Operating system cost

We will charge OEMs more for Chicago than for MS-DOS 7.

Number of distribution floppy disks

Chicago will require many more floppies than MS-DOS 7.

Hard drive size

Chicago is bigger, so a preinstall will take more time, and will require a larger hard drive.

RAM size

Chicago will require at least 4Mb of RAM. MS-DOS 7 could have a smaller RAM requirement (1Mb or 2Mb).

2.1.2. To meet End-User demand for straight MS-DOS

As RichF noted in his memo, some customers do not want Windows:

- o Don't like the GUI menus -- too complicated/confusing
- o Windows slows them down
- o Windows is too much of a resource pig
- o Windows requires more hardware than the customer has

2.2. Take every last dollar of upgrade revenue

There will be some customers in the market for an OS upgrade in 1994:

- o Customers with under powered machines (80286, or 80386 with at most 4Mb RAM).
- o Customers who think Chicago/Windows is too much for them.
- o Customers who always buy the latest OS release (but why would they buy MS-DOS 7 when they could buy Chicago?)

2.3. Competition from MS-DOS Clones

We must be prepared for Novell and IBM (and others) to be more aggressive about adding features to their products and garnering OEM business.

2.3.1. Novell DR-DOS

Key features Novell has announced (but not yet shipped) in Novell DR-DOS 7:

- o DOS Protected Mode Services -- allows drivers and TSRs to execute in 16-bit protected mode, and operate under Windows.
- o Multi-tasking of MS-DOS applications
- o Integrated Novell Client software & peer-to-peer networking

2.3.2. IBM PC-DOS

Key features IBM has announced (but not yet shipped) in PC-DOS 6.x:

- o Anti-virus (IBM internal)
- o Backup (Fastback from Fifth Generation)
- o PCMCIA 2 drivers (from Phoenix)

...

2.4. Asian acceptance of Windows is lagging U.S./Europe

This is a point paulma made, but I'm not sure I buy it. We haven't been very focused on even getting MS-DOS 6 reading in this market, and DOS5/V in Japan seems to have more momentum.

Note that if we think this is important, then we have to make sure any features we put in MS-DOS 7 can be very easily adapted for the Asian market! This would rule out taking very much of Chicago, for example!

3. Chicago *minus* the GUI

This product leaps into the future, offering customers modern, 32-bit underpinnings while retaining compatibility with MS-DOS applications.

The primary motivation is to provide multi-tasking of MS-DOS applications, and basing this on our newest technology (Chicago) seems most sensible. A different approach would be to base this on Win3.1, and skip Dragon (the 32-bit I/O system) altogether. That would remove the PNP issues, but leave us with an older code base which doesn't have the Virtual Machine improvements of Chicago.

3.1. Features

3.1.1. Multitasking of MS-DOS applications

We get most of the great features in Chicago:

- o Preemptive multitasking of MS-DOS apps
- o Big VMs (i.e., lots of free RAM)
- o All sorts of advanced VM properties (fine control over what is virtualized, how much XMS and LIM memory are available, timer tick simulation, video mode support, etc.) This entails support for the enhanced PIF format.

3.1.2. Dragon File System

The full VxD-based reentrant, protected-mode, 32-bit file system.

3.1.3. Plug-and-Play

While we may not want Plug-and-Play (see below), this will be required in order to support setting up Dragon device drivers.

3.1.4. NO long file names

No MS-DOS apps support long file names, so even if we had support in COMMAND.COM and all the other utilities, there would be very, very little end user benefit.

3.2. Work Items

3.2.1. Non-GUI Plug-and-Play

Have to make sure all of the PNP stuff works without a GUI.

Especially the boot-time detection of added/removed hardware, and user interaction to resolve situation.

3.2.2. Non-GUI Registry

PNP requires SYSTEM.DAT (and USER.DAT?) registry files, APIs, and probably a user interface for maintenance.

3.2.3. Character-mode Task Manager

RaymondC has an old one from Cougar days which could be resurrected with less than 5 days of work.

- o Start new VM

MS 0115611
CONFIDENTIAL

- o List VMs
- o Switch among VMs
- o Terminate VM
- o Program Start List (to simplify use of PIFs)

NOTE: This is designed to run in the *System VM*. Much of Windows is designed to treat the System VM as a special case, and so the path of least resistance is to maintain this situation. Thus, there is memory and processor overhead to having the Task Manager and multi-tasking.

3.2.4. Character-mode PIF Editor

There are lots of fancy new PIF settings for VMs. At a minimum, we need a PIF editor to control some minimal number of settings.

3.2.5. Character-mode Control Panel

Many system settings for the VMM and VxDs are available via the Windows Control Panel. While we would not need all of these, there are

- o Swap file location/size -- the nightmare of code and UI in Win3.1 is gone, but this is still not trivial work.
- o Device Contention settings -- COM and LPT time-out and contention resolution
- o COM port settings
- o Multi-tasking settings
- o ...

3.2.6. Setup

One approach would be to use the Chicago Setup, which includes a trimmed down version of Win3.1. Indeed, given Plug-and-Play, this is probably the only approach that would be sensible, but then we have increased the interdependence between Chicago and MS-DOS 7.

3.2.7. Enable Win3.1 to Run?

RichP has already done a prototype of this work (during Cougar) to verify it is feasible.

3.3. Disadvantages

3.3.1. Time and Team impacts to linking with Chicago

Being dependent upon Chicago would be a major, major frustration for the MS-DOS team. Chicago is clearly the more important product, so when schedules get tight MS-DOS will get pimped.

- o Chicago must be able to setup over MS-DOS 7, which is new work for Chicago.
- o There is a tremendous amount of shared code -- VMM, VxDs, Setup, USER, KERNEL, PNP, Registry, ...

3.3.2. Plug-And-Play has GUI dependencies

- o PNP Setup and Configuration is a Win3.1-level GUI application; emergency disk includes stripped down Win3.1 to run this program.
- o Hardware vendors ship VxDs and/or SCSI miniport drivers, and may ship Windows Control Panel (*.CPL) files to control special features of their drivers.
- o PNP requires USER.EXE and KERNEL.EXE to support the messaging that goes on to resolve PNP setup and configuration.
- o Since PNP drivers are designed for the full Chicago, it will be impossible to ensure that they work in the subset environment of MS-DOS 7!

3.3.3. Size and Speed Impacts

Running the VMM and VxDs requires more disk space and more RAM, and slows the system down.

- o More distribution floppies -- we have 3 right now, would have to add at least 2-3 more.
- o More hard drive space for a full installation -- VMM, VxDs, (permanent) swapfile
- o More RAM -- nothing is free
- o Decreased performance

NOTE: Task Manager will run in the System VM.

3.3.4. Reduced MS-DOS Application Compatibility

The major feature, running VMs, brings along most of the compatibility problems that Win3.1 suffers:

- o No VCPI support (have to disable multi-tasking)
- o Incompatibilities with certain video hardware
- o Timing problems affect some games, COM applications

4. MS-DOS 6 *plus* Features

This product extends the existing role that MS-DOS has played in the past, being even more parsimonious with conventional RAM, fleshing out DoubleSpace, and providing PCMCIA support. Compatibility is the key requirement for this product, since customers who buy this are not adventurous enough to go to Chicago.

4.1. Features

4.1.1. 16-bit Protected-Mode Driver Subsystem (DPMID)

Similar in concept to Novell's DPMS, this would require a 16-bit MS-DOS Extender, so that DoubleSpace, SmartDrive, etc. could load into and execute from XMS memory.

4.1.2. DoubleSpace Driver Enhancements

- o Auto-mounting of removable media
- o DPMID-aware (minimal low-memory footprint)
- o Additional robustness
- o On-the-fly MaxCompress (assuming no patent issues)

4.1.3. SmartDrive Enhancements

- o DPMID-aware (minimal low-memory footprint)

4.1.4. PCMCIA support

We need Socket Services, Card Services, Flash File System, DBLFLASH.EXE, etc.

This would be licensed from a third-party, and then we would refine and integrate it into MS-DOS 7.

4.1.5. COMMAND.COM enhancements

Take the Jaguar file find engine perhaps, maybe license 4DOS, or even write our own enhancements.

4.2. Work Items

Mostly, see the feature list above.

4.2.1. DPMI Driver Subsystem

4.2.2. DoubleSpace

4.2.3. SmartDrive

4.2.4. PCMCIA

4.2.5. Setup

4.3. Disadvantages

4.3.1. Market reaction

- o May be seen as lagging the technology in DR-DOS or PC-DOS.

5. Recommendation

MS-DOS 7 should be based on MS-DOS 6. This keeps MS-DOS 7 in the same compatibility, performance, and size arenas that MS-DOS customers are accustomed to. It avoids interdependencies between the MS-DOS and Chicago teams, which would at best slow down both product groups, and more likely result in MS-DOS getting the short end of the stick. And, it focuses on features that can be leveraged by Chicago (DPMID and PCMCIA), rather than duplicating work already being done in Chicago.

MS-DOS 7 Product Plan Strawman

Version 0.1

May 24, 1993

Richard Freedman

The purpose of this document is to initiate the discussion on future MS-DOS product plans. The opinions and suggestions expressed within are mine, and the data collected so far is admittedly sparse. This document is intended as a first step in the product planning process.

Why MS-DOS 7 at all?

Regardless of OEM penetration of Windows, there is still substantial demand for MS-DOS only on new PCs, and for that reason no one can guarantee that Chicago will penetrate 100% of new PCs. An MS-DOS product strategy that presumes any loss of OEM business- even 1% - will not be acceptable, and without an MS-DOS beyond 6 we cannot guarantee zero loss of OEM business.

But if Chicago will be a "better DOS than DOS," why develop an MS-DOS 7 at all? Why not simply license Chicago as MS-DOS? A good question with a simple answer: Regardless of how compelling the MS-DOS support is in Chicago, the main objection MS-DOS users will have is that it will still be called "Windows." People purchase MS-DOS by default because they don't like Windows.

Why people don't use Windows: What we shouldn't do in MS-DOS 7

Some recent research shows why people don't use Windows. The responses are not from random users, but rather from people with sufficient power (at least a 386SX w/4mb) who tried and rejected Windows:

Question: What problems did you have with Windows?

Response	Percentage	MS-DOS advantage over Windows
Menus too slow	35%	Familiarity of command-line interface and DOS applications interface
Took too much RAM	24%	Speed on a low-end machine? Conventional footprint in DOS box?
Slow to boot up	22%	Start-up speed
Not any easier	19%	Familiarity
Too much disk space	9%	Smaller on-disk footprint
Crashing problems	7%	Real-mode compatibility

Note that certain apps, primarily games and comm programs, run best in real mode.

Question: Why haven't you adopted Windows yet?

Response	Percentage	MS-DOS advantage over Windows
Used DOS so long no need to switch	53%	Familiarity
Windows apps offer no advantages	39%	Not related to OS
Would cost a lot to upgrade apps	35%	Not related to OS
No Windows version of apps used	29%	Not related to OS

Question: Which one or two are the most important factors in your choice not to use Windows?

Response	Percentage	MS-DOS advantage over Windows
Used DOS so long no need to switch	47%	Familiarity
Windows apps offer no advantages	17%	Not related to OS

Source: PC World Subscriber Tracking, 2/93. N=75.

Based on this information, we can outline what we shouldn't do in MS-DOS 7:

- Change the interface: No plans to do this anyway
- Increase the on-disk footprint: Since the Windows tools are going away, the footprint should shrink
- Lessen compatibility: Multitasking could be a monkey-wrench
- Penalize performance: Multitasking could be a performance hit as well, although perhaps a containable one and no worse than DoubleSpace. Also, MS-DOS will still boot at the same speed (i.e. fast), and that will greatly lessen any perceptions of slowness.

MS-DOS 7 Vision and Overview

First, several assumptions:

- Chicago will be an excellent environment in which to run DOS apps, and there is little substantive functionality we can add in MS-DOS 7 above and beyond what's in Chicago
- Users will choose MS-DOS 7 over Chicago for what it *lacks* - no GUI, less overhead - and not for what it *adds*
- Investments in MS-DOS are not leveraged since MS-DOS 7 could be the end of the line

Given these assumptions, here is a strawman vision:

MS-DOS 7 should be just good enough to justify the 7 with customers and the press, and be just good enough to fend off Novell and IBM.

In other words, we should dedicate as few resources as possible to MS-DOS 7.

Justifying the 7

We have learned several lessons from MS-DOS 5 and 6:

- To justify a major new release you need one major OS-only feature
- People will buy an upgrade for that one feature
- To satisfy customers that you have listened to them, and to get good reviews, you still need additional features. They include more ambitious "fix what's broken" features like edit, the improved shell, and backup, as well as minor fit-and-finish like DIR /S and choice

In both MS-DOS 5 and 6 we erred on the high side by throwing in everything we could get our hands on. We probably could have sold as many Upgrades and gotten as good reviews if we had, for example, not had the task swapper or anti-virus, and we might have known they were superfluous had we done more thorough research.

Predicted Competition

The major features of DR DOS 7 were pre-announced to be multitasking, protect mode drivers, and a built-in Netware client, including peer networking. Given our experience putting proprietary networking into both WfW and the MS-DOS 6 beta, it's safe to say that a proprietary peer server in MS-DOS 7 would be more of a liability than an asset.

Without having the MS-DOS source as a running start, it's unclear what IBM PSP will be able to deliver. From looking at the PC-DOS 6 feature list it appears that PSP's PC-DOS group has very limited resources. Compression (Stac), backup (CPS), anti-virus (existing IBM in-house), PCMCIA support (Phoenix), and networking (MS?) were all designed and developed outside. There appear to be no important features like DoubleSpace or MemMaker that were envisioned, designed and developed by PSP. In PC-DOS 7 some form of multitasking taken from OS/2 seems a certainty, but it's unclear if they will be able to tackle the huge task of securing a VxD server and rewriting key drivers as VxDs.

Strawman Feature Set

Customers and the press both want and are expecting architectural changes in MS-DOS 7, and we have to make a minimally acceptable effort to satisfy them. Beyond those changes, however, the list of mandatories is short. Given all the utilities we included in MS-DOS 6, nothing obvious is still missing. To limit the scope of MS-DOS 7 all we should do beyond the major features is DoubleSpace safety work and enough fit-and-finish to make people happy.

Here then is a strawman feature set:

- VxD support: The major feature
- Multitasking: Secondary but important feature, and a competitive mandatory
- DoubleSpace bullet-proofing: Obviously high priority
- Other fit-and-finish: Respond to enough customer requests like XMS diskcopy, a longer path, and copy overwrite-protection to make people happy

Evanston

The product described above would be the traditional standalone MS-DOS and MS-DOS Upgrade. Since the latter won't interest Chicago purchasers, the MS-DOS Upgrade business will implode. 82% of MS-DOS 6 Upgraders use Windows, and if one conservatively assumes that 85% of all users will upgrade to Chicago and not MS-DOS 7, then the MS-DOS Upgrade business will go from being \$600 million to \$90 million over the product lifecycle.

Furthermore, since MS-DOS 7 could be the end, \$90 million could eventually become \$0. Therefore, it seems imperative that we reverse the traditional paradigm and reinvent MS-DOS to become a Windows-based product. Should we not, we would lose the first and possibly last chance to revitalize the hugely valuable MS-DOS brand name. I would argue that we actually need three products:

- MS-DOS 7: Traditional OEM product to hold the fort
- MS-DOS 7 Upgrade: Traditional Upgrade product for the few remaining MS-DOS users
- Evanston: MS-DOS layer that sits on top of Chicago. Hence the codename "Evanston" (it also sits on top of Chicago)

Here are some strawman visions for Evanston:

Everything you love about MS-DOS plus everything you love about Windows

If Chicago is a "better MS-DOS than MS-DOS" then Chicago plus Evanston is the "ultimate MS-DOS"

The comfort of MS-DOS. The power of Windows.

Since Evanston would be the product of the future, I would argue that it should be the product that has our best talent and thought focused on overwhelming the customer. MS-DOS 7 should be the product where our concious aim is to be "just good enough."

Is there a need for Evanston?

There are several statistics of note that would seem to make Evanston plausible:

Question: How often do you use the command line or "c" prompt to type in DOS commands such as "dir" or "copy?"

Windows 3.0 users	62%
MS-DOS 5 Upgraders	75%
MS-DOS 5 OEM users	63%
MS-DOS not 5 users	62%

Source: MS-DOS / Windows satisfaction study, 4/92

At least 100 million people worldwide know how the command-line, and regardless of how much they like it, using the command-line is a skill that isn't being unlearned even by Windows users.

Question: Do you use the DOS or Windows version of the software you use most often for work? (only Windows users were in the study)

DOS apps only	31%
DOS & Win apps	45%
Win apps only	23%

Product imperatives and outstanding issues:

First, here are some suggested product imperatives for Evanston:

- It should require Chicago
- It should have features not in MS-DOS 7 and act as a carrot to entice MS-DOS-only users to Chicago
- The MS-DOS name will help Evanston rise above the Windows utilities fray, but to be clearly above the fray it must have some OS-only features
- Windows is a much better development platform than MS-DOS, and Evanston should leverage this fact

Second, there are a number of outstanding issues:

- The positioning of MS-DOS as both a standalone product and as a Windows layer could be confusing
- The continued linkage of MS-DOS to Windows could create the perception that Chicago is still based on outdated MS-DOS technology
- Customers could accuse us of gouging if they think we were arbitrary in choosing what features we put in Chicago and what others we put in Evanston
- Customers could accuse us of coercion by forcing them to buy Windows in order to get the best MS-DOS

Target customer

Evanston should be designed and targeted differently than the MS-DOS 5 and 6 Upgrades because it cannot be a "No PC should be without it" product. Since it would be an add-on and not the base OS, we should focus on the more advanced Windows user who still uses MS-DOS and/or MS-DOS apps to get work done. This focus is different than the classic MS-DOS or Windows Upgrade, which are targeted at everyone.

Although we could put some very compelling features in Evanston - and therefore target a wider audience - I fear that if we are too aggressive in bulking up Evanston at the expense of Chicago that we could both hurt Chicago and anger our customers.

Business Case

Quick numbers suggest that Evanston could be at minimum a \$90 million business:

- Based on MS-DOS 6 Upgrade sales, Chicago should sell at least 10 million upgrades
- The Font Pak model (sharp focus on selling one cheap, related add-on) seems to work much better than the Upgrade Your World model (a broad push to sell many unrelated products)
- The Font Pak attach rate was 50% on direct sales; assume a focused add-on campaign in the channel could achieve 10% attachment
- Assume that if Evanston attaches to enough Chicago Upgrades at launch, it will gain its own momentum and sustain ongoing sales equivalent to another 10% of Chicago Upgrade volume

$$10,000,000 * (10\% + 10\%) * \$45 = \$90,000,000$$

I have not estimated Evanston purchases on OEM Chicago systems or possible Evanston OEM revenues.

Strawman Feature Set

As a baseline, the new MS-DOS features in Chicago will be (according to jeffpar and russ arun):

- Zero footprint: DoubleSpace VxD, Mouse VxD, VFAT and zero footprint Smartdrv, MSCDEX VxD
- Long file names (won't pass through to DOS apps unless they are updated)
- Toolbar for windowed DOS apps
- TrueType support in windowed DOS boxes
- Files high, Lastdrive high
- Launch Win apps from the command-line
- 1024 cylinder support
- Last access date
- Better support for graphics mode DOS apps in windowed DOS boxes

MS-DOS features that are definitely not in Chicago, or are under debate:

- Johnhe's command-line find engine
- Ability to load device drivers from the command-line
- 255 char path - undecided
- 736K DOS boxes
- Flexboot multiple OSes - undecided
- A DOS PIF setting that allows a batch file to execute before an app (local configuration)

Here are some possible feature areas for Evanston to help make the idea more concrete:

Super command-line - "The best of CUI and GUI"

We know that Windows users use the command-line, and we know from CompuServe data that advanced users are passionate about it. The command-line could stand a lot of improvement, and in Windows it's not easily accessible. We could also add many features in a Windows command-line, like color, that would be hard to add in MS-DOS. Here are some ideas:

- Unlimited scroll-back
- Split-screen support
- Font and color control, including color control for different file attributes
- Right-mouse click brings up last X commands and common commands menu (dir, copy, etc.)
- Ability to run Win apps from the command line (already in Chicago)
- Ability to can Progman/Fileman/Explorer and use the command prompt as the Windows shell
- Optional button bar for common commands
- Complete cut-and-paste support
- Choice between TTY error messages and alert box error messages
- Customizable C: prompt and C: prompt editor
- Forgiving parser for non-destructive commands - "dor *.txt" would execute as "dir *.txt"
- Improved error messages and error recovery (Paul Somerson's Wizard concept)
- Dir files, then highlighting them with mouse and pressing DEL deletes them
- Built in DOSKEY

Command-line command enhancements

- Command w/o args brings up command dialog box (copy, xcopy, format, etc.)
- "The Copy Monster" - XCOPY and DISKCOPY in one, XMS support, prompt for new disk when full, file spanning, dialog boxes with help if no args

Improved memory management

- 736K DOS boxes for text mode apps
- Additional VxDs not in Chicago: UNDELETE, RAMDRIVE, etc.
- Load device drivers from the command-line plus a DOS PIF setting that allows a batch file to execute before an app equals local configuration control by app

Other random ideas

- Windows disk tools - diskfix, background defrag, Aaron's magic tool
- Flexboot if Chicago doesn't do it
- Long path if Chicago doesn't do it
- "Install as MS-DOS-only" - would delete all the Windows-specific files (applets, wallpaper, etc.)
- Take Windows AV and Backup from Chicago. Apparently Chicago uses backup to make floppy sets from pre-installed systems, and so this maybe impossible
- Improved batch language

Appendix A - CompuServe Beta Forum Research

Andy Thomas has tabulated all of the responses to our posting asking for MS-DOS 7 suggestions on our beta forum. There is obviously a huge power-user bias in the sample. The top individual requests were:

Longer filenames	32
New file system	24
Get rid of 640k barrier	24
32-bit	22
Tape backup support	18
One-pass diskcopy	15
peer-to-peer networking	10
Pre-emptive multitasking	9
Enhanced batch language	9
Dialog box-style utilities	8
Longer path	7
Break 1024 cylinder barrier	7

I also read all the threads individually to get a qualitative feel, and not surprisingly the overwhelming request is for a "modern OS," i.e. a 32-bit, protect-mode OS. People aren't clear about exactly what they want when they say "32-bit protect mode," but I think most would be happy with just long file names and the end of conventional memory management.

But beyond specific feature requests, I wanted a sense of people's general areas of interest. Towards this end, I grouped all the requests into 7 categories. Note that people are extremely interested in an improved command-line and command set.

Category	#	Description	Examples
Architectural enhancements	11	Any suggestions to the effect of "I want a new OS"	Long file names, New file system, 32-bitness, multitasking, 1024 cylinders
	2		
Command environment	75	Suggestions related to activities performed from the C>, including command-line utilities	One-pass diskcopy, dialog box-style utilities, longer path, colors without ANSISYS, "use 4DOS"
Utilities	69	Suggested enhancements to existing utilities or new non-command-line utilities	Tape backup, improve the shell
Memory management	46	Suggestions specifically related to memory management	Break to 640K barrier, DPMS support
Other	18	Assorted random stuff	Security features
Networking	16	Self explanatory	Peer networking
DoubleSpace	16	Self explanatory	Better interoperability with Windows
Batch language	14	Self explanatory	"Improve it," return more errorlevels