

**Portable Systems Group**

**NT OS/2 Design Workbook Introduction**

**Author:** *Lou Perazzoli*

*Original Draft 1.0, March 31, 1989*

*Revision 2.0, May 5, 1989*

*Revision 3.0, August 17, 1989*

*Revision 4.0, October 15, 1989*

*Revision 5.0, January 15, 1990*

*Revision 6.0, July 25, 1990*

***Hardcopy released to The Smithsonian Institute.***

***Digital copy released to Universities for non-commercial academic use under the Windows Research Kernel License.***

1. Introduction.....	1
2. Project Goals.....	1
3. NT OS/2 Components.....	2
4. Functional Specifications.....	3

## 1. Introduction

The **NT OS/2** system is a portable implementation of OS/2 developed in a high-level language. The initial release of **NT OS/2** is targeted for Intel 860-based hardware, which includes both personal computers (*Frazzle*) and servers (*Dazzle*).

The first systems based on a RISC microprocessor will be available for testing in the fall of 1990.

## 2. Project Goals

The ultimate goal of the **NT OS/2** project is to develop a portable implementation of OS/2 executing on the Intel 860 and to establish this combination of hardware and software as the standard for high-performance personal computers and server systems.

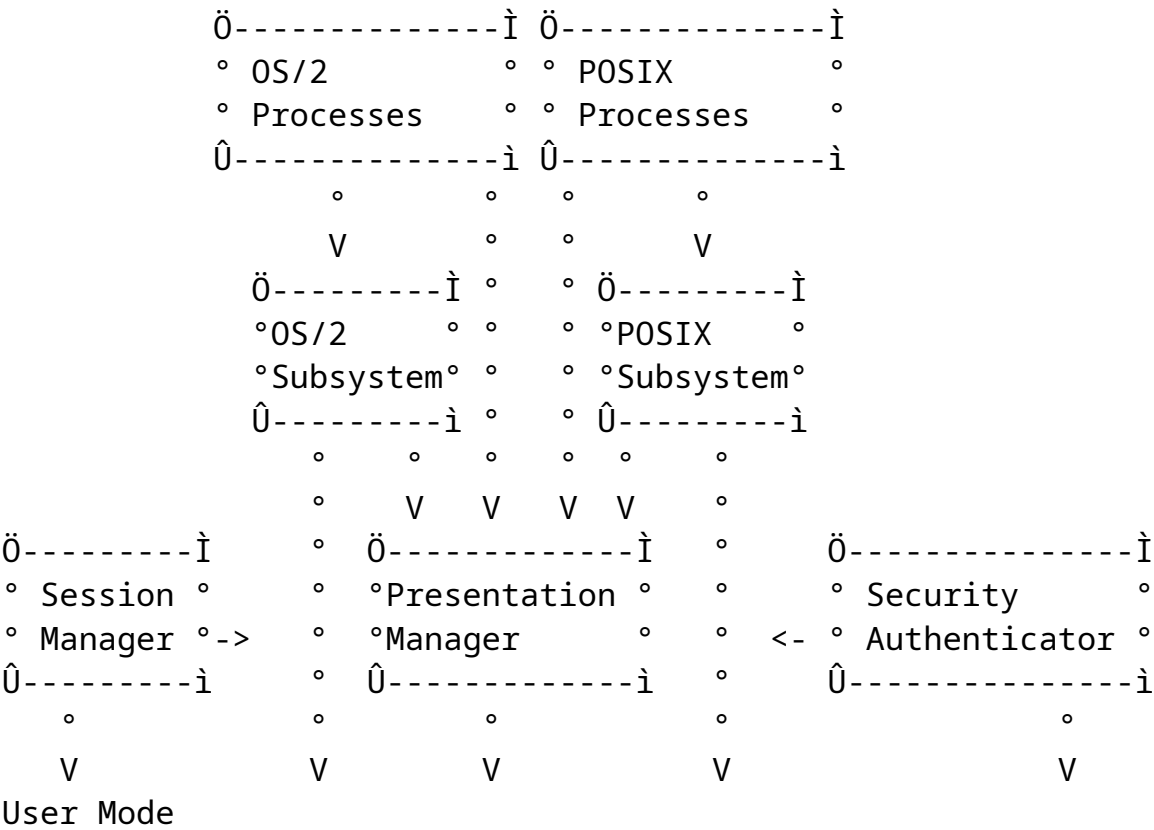
**NT OS/2** has the following overall project goals (though not all these goals will be attained by the first implementation of **NT OS/2**):

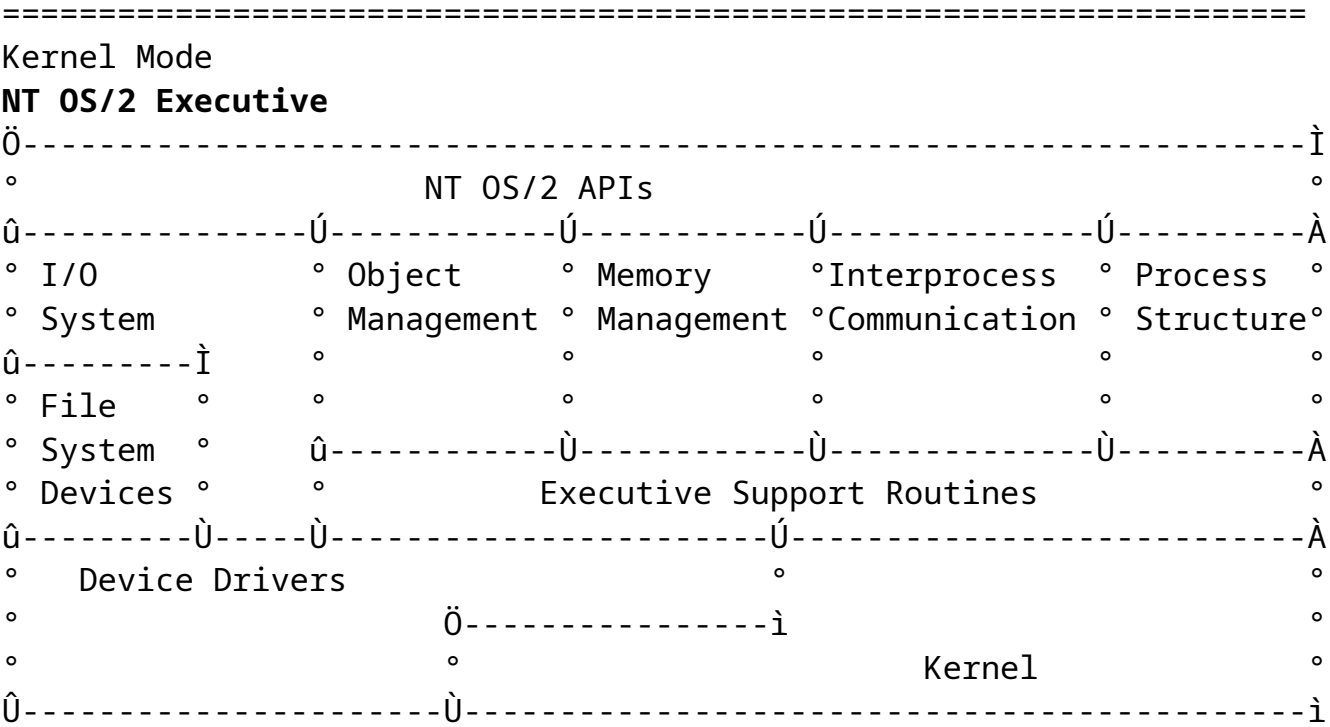
- o Portability to a variety of hardware architectures. Though the first implementation is targeted to the Intel 860, the overall system design isolates the machine-dependent portions for portability to other architectures.
- o Support for multiple processors with shared memory via symmetric multiprocessing. This provides performance improvements for multiprocessor workstations and servers.
- o Compatibility with the OS/2 V2.0 32-bit application programming interface (API). Because the initial target system is not an Intel x86 architecture, all applications will have to be recompiled and relinked. In addition, any assembly language code will have to be rewritten or converted to a higher level language such as C.
- o Security at the C2 level with future versions achieving higher levels of security. This includes login/logout options on the personal computer and the server system, and declaration and enforcement of protection attributes for shareable resources (files, IPC, memory objects, etc.).
- o Support for a POSIX-compliant API interface that passes the POSIX validation suites.

- o Support for internationalization.
- o Support for LANMAN networking and management of personal computers and servers.
- o Support for the current Presentation Manager API running in both the OS/2 environment and the POSIX environment.
- o Support for distributed applications. The network is integrated into the system to allow transparent distribution of applications and services within a network.
- o Support for object-oriented file systems and object-oriented presentation manager.
- o Easy extensibility by layering new features on the existing system without modifying the underlying system.
- o Simultaneous execution by multiple users, each with a unique security profile.
- o Interoperability and data interchange between OS/2 and POSIX applications.
- o High reliability that prevents errant user programs from causing a system crash or exhausting system-wide resources. Resource quotas, a protected kernel, and protected objects are used to improve reliability.

### 3. NT OS/2 Components

**NT OS/2** consists of a highly integrated kernel / executive that executes in kernel mode. It provides the necessary services to allow the emulation of OS/2 and POSIX APIs via protected subsystems executing in user mode. Both the OS/2 and POSIX subsystems provide these services through remote procedure calls from a client to the server subsystem. The server subsystem, in turn, emulates the desired operation locally or by calling the executive, and returns the results to the caller. The following diagram illustrates the structure of **NT OS/2**.





Block Diagram of NT OS/2:

4. Functional Specifications

The following specifications are contained within this design workbook. Each specification contains an abstract of the component it describes, how that component fits into the system, the various APIs that are used to access the functionality, and enough detail to ensure the defined capability can be implemented.

The goal of the specifications is to allow someone to understand the functionality provided by a particular piece of the system. It is NOT a goal to describe the actual implementation.

Each specification addresses Cruiser and POSIX compatibility, if appropriate. The following is a list of design specifications included in this version of the workbook:

- 1. Kernel - Describes the function of the kernel, the objects implemented, and the various interfaces provided to manipulate these objects. This specification contains implementation details, where necessary, to reveal how

multiprocessing and processor dispatching take place. This specification also describes synchronization, scheduling/dispatching, and Asynchronous Procedure Calls (APCs).

2. Object Management - Describes how the executive deals with objects, what they are for, how they are protected, how they are named, how they are allocated, how they are accounted for, and how they are deleted. This specification also addresses object directories and how to access them using the file system directory operations.
3. Process Structure - Describes the process and thread objects and the operations that can be performed on them. This specification also explains signals and how OS/2 compatibility and POSIX compliance are addressed.
4. Virtual Memory - Describes the virtual memory objects and the operations that can be performed on these objects.
5. I/O Management - Describes the APIs and objects available for I/O operations.
6. Security - Describes how security is provided in the system, the ACL format, ACL access checking rules, login/logout, the authorization file, and the partial closure of covert channels. This specification also describes audit and alarm logging.
7. Local Process Communication - Describes the client/server protected subsystem model, client impersonation, port objects, and connection/disconnection operations.
8. Remote Procedure Call - Describes a transport-independent interface to remote procedure calls.
9. Session Manager - Describes how the subsystems for OS/2 and POSIX are created, and how they interact with each other.
10. File System - Describes the file systems, how they are put together, the functions they perform, and how they accomplish the tasks that they are given.
11. Semaphores and Events - Describes the APIs and objects available for synchronization.

12. Argument Validation - Describes the argument probing and capture requirements for system services.
13. Timers - Describes the timer object, which is used to mark time, and the functions available to manipulate it.
14. Coding Guidelines - Describes the naming and structure of **NT OS/2** code.
15. LAN Manager Software - Describes the network capabilities of the system, how network drivers fit together, and how the protocol stacks are managed.
16. Exceptions - Describes the dispatching of hardware exceptions to the condition dispatcher and the arguments that accompany each exception. It also explains guard page handling, automatic stack expansion, and access violations on the user stack, as well as how signals are handled at the user level.
17. OS/2 Emulation Subsystem - Describes the requirements and methods used to design and build the OS/2 emulation subsystem.
18. Status values - Describes the format for status values return by **NT OS/2** APIs.
19. Subsystem Design Rational - Describes the rationale for designing OS/2 and POSIX emulation as subsystems as opposed to supporting the APIs directly in the executive.
20. Shared Resource Specification - Describes the routines that implement multiple-readers, single-writer access to a share resource.
21. Executive Support Routines - Describes executive support routines which are available in kernel mode and not documented in other chapters.
22. Driver Model - Describes the device driver model, how I/O is managed throughout the system, how the file system and network capabilities fit into the system, and the objects and operations that are available to help manage the I/O system. It also presents I/O validation, queueing, page lockdown, double mapping, I/O completion, and error logging.
23. POSIX Emulation Subsystem - Describes the requirements and methods used to design and build the POSIX emulation subsystem.



24. Time Conversion Specification - Describes the APIs available for viewing time and converting to and from different formats.
25. Mutant Specification - Describes the mutant object and services which operate upon the object.
26. Transport Driver Interface - Describes the interface for the network transport layer.
27. Network Driver Interface Specification - Describes the interface for the network physical layer.
28. Lan Manager Server - Describes the design of the Lan Manager server and the operations supported.
29. C Structured Exception Handling - Describes the extensions to C in the MS 860 compiler to support structured exception handling.
30. NT C User's Guide - Describes the command syntax and language issues for the MS 860 C compiler.
31. Prefix Table - Describes the prefix table package.
32. System Startup Design Note - Describes system startup after phase one initialization.
33. Debug Architecture - Describes the debug architecture for NT OS/2.
34. Linker/Librarian - Describes the NT OS/2 linker, librarian, and image format.
35. Caching Design Note - Describes the system-wide file caching implementation.
36. Utility Design Specification - Describes the basic support routines for NT OS/2 utilities.
37. OS/2 Environment Subsection Security - Describes the security features of the OS/2 environment subsystem.
38. Security Account Manager Protected Server - Describes the security account manager which maintains user and group account information.

**Revision History:**

Original Draft 1.0, March 31, 1989

Revision 2.0, May 5, 1989

1. The following specifications were added to the design workbook:
  - o Local Process Communication
  - o File Systems
  - o Session Manager
  - o Semaphores and Events
  - o Argument Validation
  - o Timers
  - o Coding Guidelines
2. The in-progress specification list was changed to add the OS Emulation Environment specification.
3. The block diagram was modified.

Revision 3.0, August 17, 1989

1. The following specifications were added to the design workbook:
  - o Subsystem Design Rationale
  - o Status Codes
  - o Shared Resources
  - o Executive Support Routines
  - o User-mode Interlocked and Fast Lock Routines
  - o OS/2 Subsystem Emulation

Revision 4.0, October 15, 1989

1. The following specifications were added to the design workbook:
  - o POSIX Subsystem Emulation
  - o Time Conversion Specification
2. The User-mode Interlocked and Fast Lock Routines Specification was dropped because the APIs were non-portable.
3. The specification list was revised to match the actual specifications in the workbook.
4. The File System Specification was replaced by the File System Design Note.
5. The I/O System Specification was broken into two separate specifications:
  - o I/O System Specification - Documents I/O system API
  - o Driver Model Specification - Documents drivers and I/O system internals

Revision 5.0, January 15, 1990.

1. The following specifications were added to the design workbook:
  - o Mutant Specification
  - o Transport Driver Interface
  - o Physical Driver Interface
  - o Lan Manager Server
  - o C Structured Exception Handling
  - o NT C User's Guide

Revision 6.0, July 25, 1990.

1. The following specifications were added to the design workbook:
  - o System Startup Design Note

- o Debug Architecture
- o OS/2 Linker/Librarian/Image Format Specification
- o Caching Design Note
- o Utility Design Specification
- o OS/2 Environment Subsystem Security
- o Security Account Manager Protected Server