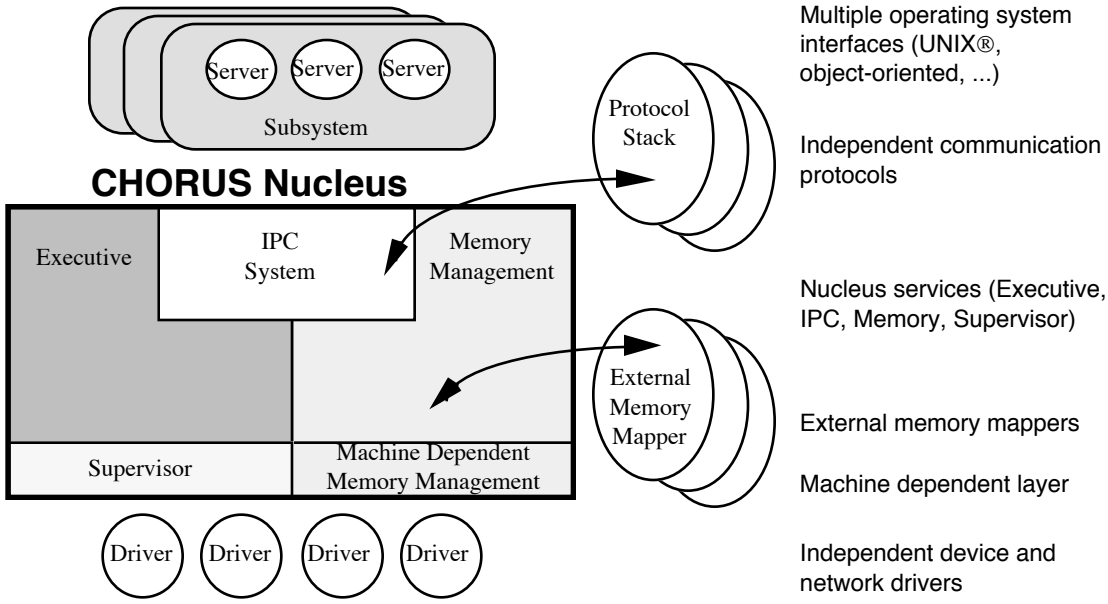
Product Datasheet: CHORUS® Nucleus v3 r4.2

- ❖**Microkernel for operating systems**
- ❖**Supports multiple-server and single-server operating system architectures**
- ❖**Provides support for UNIX®-based, object-oriented and other operating systems**
- ❖**Multiple concurrent threads per actor (process)**
- ❖**(A)symmetric shared memory multi-processor hardware support**
- ❖**Real-time functionality and behavior**
- ❖**Fast, transparent location-independent message-passing system**
- ❖**Provides both system space and user space actors (processes)**
- ❖**Communication protocols, drivers and memory mappers external to Nucleus**

The CHORUS® Nucleus v3 is the heart of the CHORUS family of real-time distributed operating system products, notably the CHORUS/MiX™ UNIX®-based operating systems. The Nucleus provides a unified virtual machine interface for various operating system (subsystem) implementations. Each subsystem implements one operating system interface by using the Nucleus services. Full low-level services are provided to subsystems by the Nucleus working in association with independent device drivers, network drivers, communication protocols, and memory mappers (paging, swapping, etc.).



The product of over thirteen years of refinement and in its fourth generation, the Nucleus v3 is an extremely efficient, compact and modular microkernel with a stable interface. With the release in 1989 of CHORUS/MiX V.3.2, the Nucleus was the first microkernel supporting commercial multi-server operating systems with performance equaling or bettering native monolithic operating systems. Performance-tuned to ensure responsiveness and fast Inter-Process Communication, the Nucleus integrates these essential services:

- A real-time executive synchronizes and preemptively schedules priority-based tasks.
- Modern, optimized Inter-Process Communication (asynchronous IPC and Remote Procedure Call (RPC)) mechanisms give efficient, transparent access to resources distributed on a network or bus.
- A low-level hardware supervisor dynamically loads interrupt and trap handlers.
- An advanced (optional) distributed virtual memory manager optimizes distributed object copying.
- Support for symmetric and asymmetric shared memory multi-processors.

Main Benefits

- Rapid real-time response to external events
- Simple, dynamic migration, evolution and reconfiguration of systems and servers
- Easier, transparent distributed programming with automatic localization
- Fast, efficient local and remote RPC and IPC
- Contains very short critical sections for maximum interruptibility
- Increased continuity of service via enabling technologies for fault-tolerant servers
- Enabling technologies for secure systems
- Supports multiple operating systems ("personalities") simultaneously
- (A)symmetric shared memory multi-processing
- Supports both system space and user space actors
- Dynamically loadable device drivers, not embedded in the microkernel
- Configurable kernel from 100 Kbytes static size to 250 Kbytes

Real-time Executive

- Light-weight threads (tasks/actors)
- Preemptive priority-based scheduler
- 256 priority levels, FIFO within levels
- Multiple scheduling classes
- Synchronization by events, mutexes, semaphores and mini-messages
- Execution control ("suspend", "resume")
- Separate address spaces (optional)
- Time-outs (optional)
- Allocation/deallocation of memory
- Dynamic kernel resource management
- 11 microseconds Interrupt Latency on Compaq 486 at 50 Mhz
- 22 microseconds Context Switch on Compaq 486 at 50 Mhz

Inter-Process Communication

- Fast, distribution-transparent IPC and RPC
- Optimized local and distant communication
- Small untyped messages for fastest processing by network interfaces
- Efficient low-overhead internal protocols
- Automatic localization of distributed objects providing transparent remote access to distant objects as if they were local
- A high level network interface independent of specific network protocols
- Supports standard or private network protocols
- Supports multiple subnets, each with its own network protocol
- Various facilities for managing and broadcasting to groups of destinations
- At-most-once RPC semantics
- Unique "light-weight RPC" performance optimization
- Abort propagation through RPC's

Distributed Virtual Memory Management

- Optional: deconfigurable for real-time use
- Demand-paged possibly over network so pages are brought into memory only when needed
- Separate, protected address spaces
- Can share memory between different tasks
- Supports linear, protected (segmented) and paged virtual memory

Hardware Supervisor

- Dynamic (de)install of interrupt, trap handlers
- User-level handlers installable in system space
- Multiple priority-ordered handlers
- Dynamic resource allocation and management

Validation

- Comprehensive suite of specially designed tests for exercising both local and remote operations
- Execution of CHORUS/MiX V.3.2 and/or CHORUS/MiX V.4 validation tests (see corresponding data sheets)

CHORUS Nucleus Calls

actorCreate
actorDebugged
actorDelete
actorExcept
actorPi
actorPriority
actorSelf
actorStart
actorStop
chorusCtx
chorusPTimer
chorusSite
chorusTime
chorusTimeL
chorusTod
grpAllocate
grpPortInsert
grpPortRemove
ipcCall
ipcGetData
ipcReceive
ipcReply
ipcRestore
ipcSave
ipcSend
ipcSysInfo
ipcTarget
lcFlush
lcSetRights
lcStat
mpInit
mpReceive
mpSend
mutexGet
mutexInit
mutexRel
mutexTry
portCreate
portDeclare
portDelete
portDisable
portEnable
portLi
portMigrate
portPi
portUi
rgnAllocate
rgnDup
rgnFree
rgnInit
rgnInitFromActor
rgnMap
rgnMapFromActor
rgnSetInherit
rgnSetOpaque
rgnSetPaging
rgnSetProtect
rgnStat
semPnAbort
semInit
semP
semV
sgFlush
sgRead
sgStat
sgSyncAll
sgWrite
svAbortHandler
svCallConnect
svCallDisConnect
svCheckUserSpace
svCopyIn
svCopyOut
svExcHandler
svItConnect
svItDisConnect
svLock
svLockInit
svMaskAll
svMaskedLockGet
svMaskedLockRel
svMsgHandler
svMsgHdlReply
svPagesAllocate
svPagesFree
svSpinLockGet
svSpinLockRel
svSpinLockTry
svTimeOut
svTimeOutL
svTrapConnect
svUnLock
svTrapDisConnect
svUnCTimeOut
svUnMask
svUnMaskAll
svUnTimeOut
threadAbort
threadAborted
threadBind
threadContext
threadCreate
threadDelay
threadDelete
threadList
threadLoadR
threadPriority
threadResume
threadScheduler
threadSelf
threadStoreR
threadSuspend
threadTimes
uiBuild
uiClear
uiCreate
uiDeclare
uiForget
uiIsLocal
uiSite
uiValid
vmCopy
vmFlush
vmFree
vmLock
vmPageSize
vmStat
vmPhysAddr
vmSetPar
vmUnLock

Memory Mapper Calls

MpCreate
MpGetAccess
MpPullIn
MpPushOut
MpRelease

CHORUS Nucleus Deliverables

The license fee for the CHORUS Nucleus reference port includes delivery of the following items.

Nucleus Deliverables

CHORUS Nucleus, including the following major components:

- Real-time Executive
- Virtual Memory (VM)
- Inter-Process Communication (IPC)
- Supervisor
- Networking protocols, Ethernet CHORUS Network Protocol

System-dependent code:

- Booting/initialization
- Port to the Intel i386® and Intel i486®, COMPAQ® Deskpro and Portables family
- Network driver: Excelan/Federal Tech Exos 205T, Western Digital 80x3

Include files and libraries for the above

Documentation Deliverables

Programmer's Reference Manual:

- Nucleus Specification and Interface
- Nucleus Programmer's Reference Manual
- Nucleus From r4.0 to r4.2
- Nucleus Kernel Debugger User's Manual for COMPAQ Deskpro

Implementation Guide:

- Kernel Implementation Guide
- Kernel Network Architecture
- Kernel Network Devices Manager
- Source Installation Guide

Build Environment Deliverables

- Makefiles
- Test programs
- Porting utilities
- Validation tests
- Makedepend (determines file dependencies)
- Uu (encode/decode tools)

Pre-requisites

- Standard "C" development environment (make, "C", ...)
- Compilers: Glockenspiel C++ version 2.0A
- A machine capable of reading QIC streamer tapes

Recommended:

- COMPAQ reference machine (see separate configuration information)

Media

- Source-code products: 60 MB QIC-24 streamer cartridge tape

Reference Machine

- COMPAQ 386/486

Additional Ports

- Motorola 680x0
- Sun SPARC
- MIPS R3000
- Inmos Transputer T425/T805

CHORUS is a registered trademark of Chorus Systems.
CHORUS/MiX is a trademark of Chorus Systems.
UNIX is a registered trademark of Unix System Laboratories, a wholly-owned subsidiary of Novell, Inc.
All other trademarks mentioned belong to their respective owners.
Specifications are subject to change without notice.
©1989-1994, Chorus Systems.
CSI/MR-94-1. Printed in USA.