THE POSSIBILITIES ARE INFINITE FUJITSU

WHITE PAPER

POSIX in BS2000/OSD

POSIX links BS2000/OSD and UNIX-based system environments

Issue November 2012

Pages 16

POSIX, the Portable Open System Interface for UNIX, comprises a series of standard interfaces available under UNIX-based systems.

The POSIX subsystem in BS2000/OSD enables UNIX applications to run on a BS2000/OSD mainframe without problems after they have been ported, making it possible to use many popular UNIX applications, especially those relating to internet, on a BS2000/OSD server.

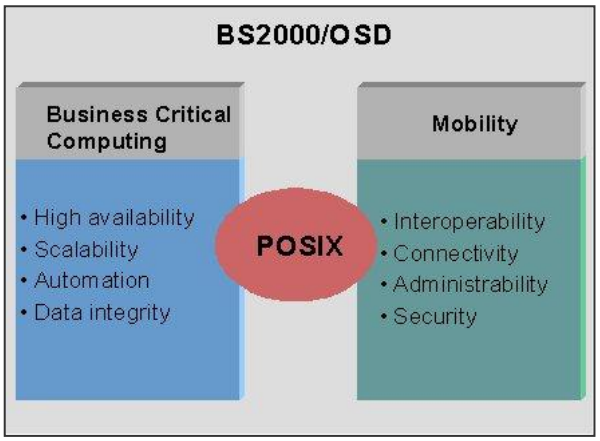
Contents

The benefits of portability and interoperability

Business-critical computing and mobility

Modern IT systems are expected to deliver maximum security, reliability, scalability and performance while still being open enough to ensure high levels of flexibility and straightforward integration of all business processes. BS2000/OSD fulfills these requirements through its strategic commitment to business-critical computing (BCC) and mobility. Mobility in this context means giving IT users access to applications from anywhere and at any time. The performance provided through this access is assured and maximum use is made of the necessary IT infrastructure.

Business-critical computing means that all data and applications are available to the participants in the business process at all times.



Open, POSIX-based applications benefit from the business-critical computing properties of BS2000/OSD. These properties include high availability, great scalability, mass throughput of data and highly cost-effective operation in data centers. POSIX also serves as the technical platform for implementing services that enable interoperability and connectivity between BS2000/OSD and the open systems world; the entire BS2000/OSD system can in fact be expanded into a full internet Server/OSD using POSIX.

A large number of widely-used commercial products, including many applications favored in internet environment, that were previously only available for UNIX systems are now available for BS2000/OSD thanks to POSIX.

POSIX opens up a wealth of new possibilities for the BS2000/OSD user including:
- easy use of internet protocols;
- the ability to set up web and mail servers on a BS2000/OSD platform; and
- the opportunity to use commercial and scientific UNIX applications.

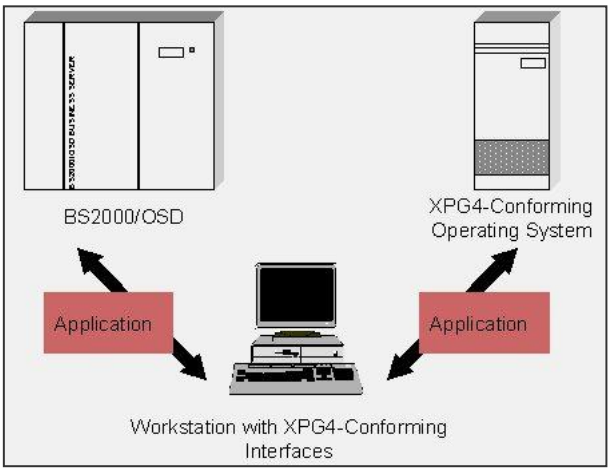
Portability and interoperability: definitive user benefits

Open systems facilitate portability and interoperability. These primary attributes of open systems create a range of benefits for the user.

Portability

Application programs that have been written for the POSIX interfaces can be executed on all XPG4-compliant operating systems and hardware platforms:

- Applications from software vendors or UNIX applications created in-house can be ported to and used on BS2000/OSD systems.
- When creating new applications, customers can write especially for the XPG4 interfaces and thereby gain long-term protection for their investment. International standardization and popularity ensure that these interfaces will remain stable for a long time to come.
- Customers can enhance existing software by adding XPG4 interfaces. BS2000/OSD and POSIX program interfaces are compatible and can be called in the same program.



Interoperability

All application programs running on operating systems that conform to the XPG4 standard can exchange data with each other. Comprehensive copy and conversion routines are available to support the reciprocal use of BS2000/OSD and POSIX text files.

The POSIX system

Structure, functions and standards

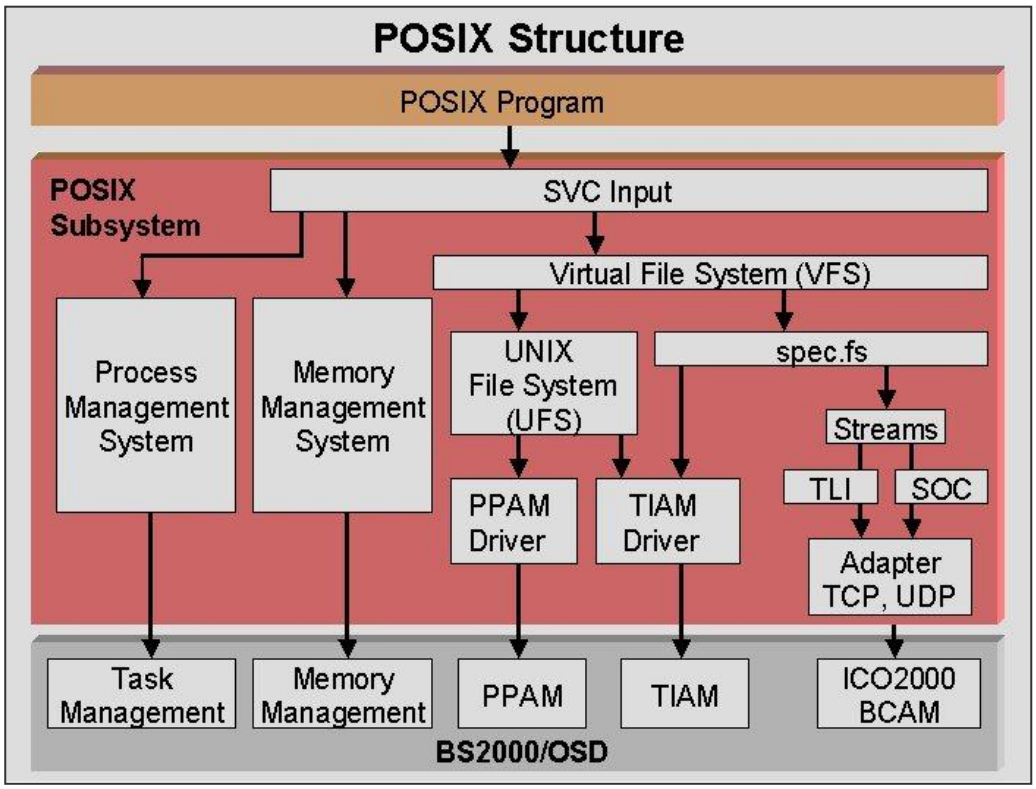
The POSIX standard was defined by the Institute of Electrical and Electronics Engineers (IEEE) in 1989 as a US national standard. It was then extended by the X/OPEN consortium and approved in 1990 as an international standard. POSIX functionality was integrated into BS2000/OSD so that these standards could also be used on BS2000/OSD mainframe. POSIX successfully gained certification in two stages: BS2000/OSD was awarded the XPG4 Base brand (XPG4) by The Open Group (formerly X/OPEN) at the end of 1995 and received the XPG4 UNIX Profile brand (also called XPG4.2 or UNIX95) in mid-1997.

How POSIX is embedded in BS2000/OSD

POSIX is a privileged BS2000/OSD subsystem that processes requests from privileged and non-privileged users. POSIX is incorporated perfectly into BS2000/OSD architecture and makes use of the resources of the basic system, the data management system and the data communication system.

The POSIX subsystem consists essentially of three parts:
- a UNIX system kernel that has been ported to BS2000/OSD;
- BS2000/OSD connectors and services that provide a link between the ported UNIX system kernel and BS2000/OSD; and
- routines for initializing and terminating the POSIX subsystem.

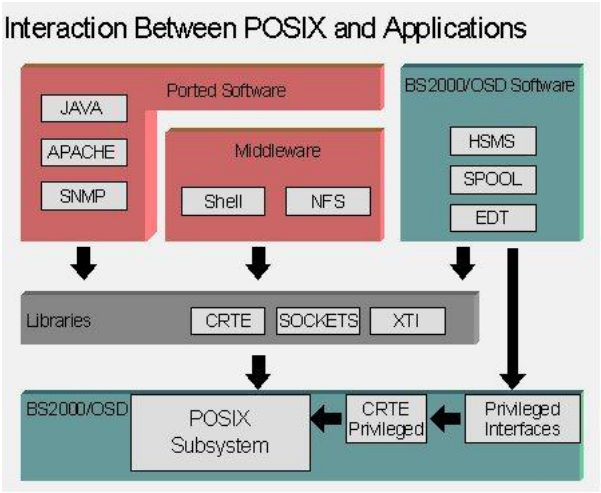
The POSIX subsystem supports the POSIX file system. Programs calling POSIX functions are directed via a special SVC to the POSIX subsystem, with subsequent branching according to the POSIX function being called. The subsystem itself is a ported UNIX system kernel with associated components such as process management, memory management, file management, etc. These components are linked via special adapters to the corresponding components of the BS2000-CP element. A POSIX file system, for example, is implemented here by means of a BS2000/OSD-PAM file hard disk and access in the POSIX subsystem to this file system is implemented via drivers that use the interfaces of the PPAM BS2000 disk driver. Access from the network is handled similarly, with the ICO2000 BS2000 interface of the BCAM BS2000 network driver being used. The privileged POSIX subsystem is started and terminated using interfaces of the DSSM subsystem manager.



Interaction of POSIX and BS2000/OSD

POSIX and BS2000/OSD interact via the program interfaces and the shell. Access to POSIX is implemented via libraries such as CRTE and Sockets. The applications use these interfaces. A distinction may be drawn between the following types of POSIX applications.

- **Standard applications**
  These include Shell and NFS along with other applications usually used under UNIX systems.
- **Applications newly ported to POSIX**
  Recently ported applications include Java, APACHE and SNMP, among others (see page 11).
- **BS2000/OSD applications that use POSIX interfaces**
  A number of BS2000/OSD applications make use of the POSIX interfaces. One example is EDT, which supports POSIX files as well as BS2000/OSD files. These BS2000/OSD programs also require privileged status to run in some cases, so a portion of the CRTE standard library is also made available in privileged mode.

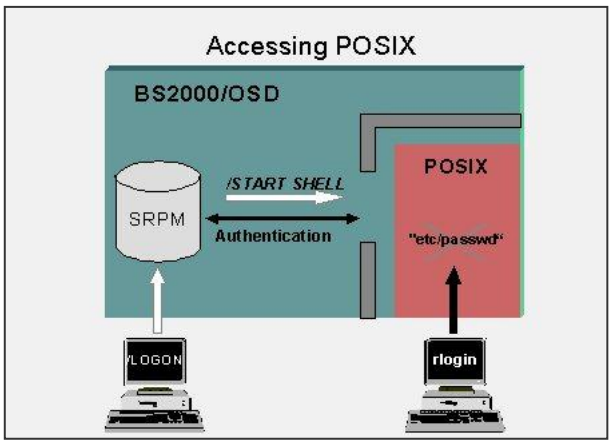


It is possible to use either the POSIX shell or the command level of BS2000/OSD when operating block terminals. Proper interaction between POSIX and BS2000/OSD is assured in both cases.

POSIX command interfaces

The POSIX shell is a command interface that can be used in addition to the command interface of BS2000/OSD. The user has no direct access to the kernel and can issue commands to POSIX only via the POSIX shell. The POSIX shell thus represents the link between the POSIX user and the POSIX kernel. The POSIX shell features a rich command language that can be used like a procedure language.

The POSIX shell can be accessed as follows:
- via a BS2000/OSD terminal (block terminal);
- from a UNIX-character terminal or a Windows-character terminal emulation on the PC, via rlogin, telnet or ssh.



Commands for a POSIX shell can be entered on a terminal or read from a file. A file containing command procedures for the shell is called a shell script. A shell script is called in the same way as a command. Once called, the commands in the shell script are executed. Shell scripts can also be called by applications.

A POSIX shell is started in BS2000/OSD using the /START-POSIX-SHELL command. This makes the shell environment available. Users leave the shell with the *exit* command. Other commands to support interaction between BS2000/OSD and POSIX are available in addition to the standard commands. These include copy functions *(bs2cp*) and BS2000/OSD commands *(bs2cmd)* that can be called directly from the shell.

The SDF command /EXEC-POSIX-CMD enables users to call POSIX shell commands, both individually and in sequence (scripts), from BS2000. Also the output of forked commands is provided on SYSOUT, so a complete protocol is guaranteed. Complete symmetry has thus been achieved: commands can be issued from the BS2000 command level to the POSIX command level and vice-versa and files can be copied back and forth without leaving the current level.

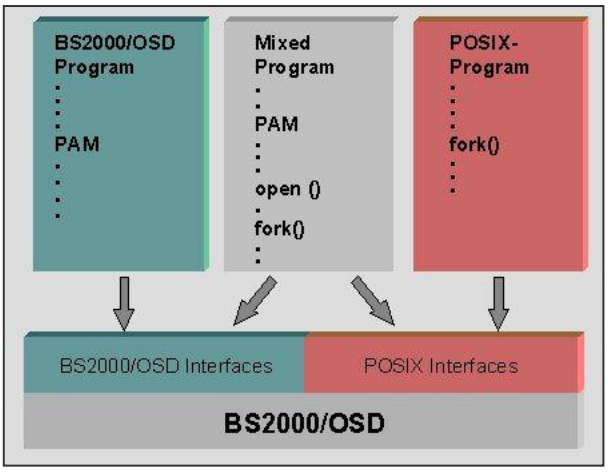
A global program/command cache is provided with the *posdbl* command in order to enhance performance. User-specific caches can be set up using the *pdbl* program.

POSIX program interfaces

The POSIX program interfaces are provided in a library that can be used by C programs. POSIX programs can be started from BS2000/OSD as well as POSIX. The POSIX program interfaces are available together with the BS2000/OSD program interfaces. Pure BS2000/OSD programs, pure POSIX programs and mixed programs can be executed. POSIX programs can be stored in both BS2000/OSD files and POSIX files. The compilers for C and C++ can be called from BS2000/OSD and from the POSIX shell.

A range of interfaces commonly used in the UNIX system world are released in an “add-on lib” in addition to the standard interfaces. Additional interfaces for using shared libraries are also available.

The most important POSIX interfaces are also offered in privileged mode to assist system-support software wishing to make use of the POSIX functions from within the system kernel.



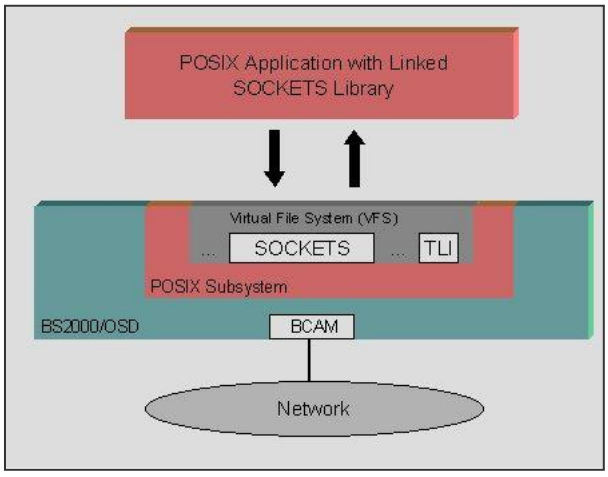
POSIX network interfaces

Sockets

The POSIX subsystem also includes the Sockets standard, a collection of client/server-based interfaces for network programming. These interfaces are described in SPEC 1170, which is an expansion of XPG4. They enable access to the internet via TCP/IP and UDP/IP, with access to the internet representing access to the “open network world”. A distinction is drawn between communication based on TCP/IP and communication based on UDP/IP. These communication types are provided with the Sockets types SOCK_STREAM (TCP) and SOCK_DGRAM (UDP) respectively. Stream sockets (TCP) implement connection-oriented communication, which establishes a permanent connection between two communication terminals. The transmitted data packets arrive at the receiver in the same order in which they left the sender. Connectionless communication is implemented using datagram Sockets (UDP).

Stream Sockets and datagram Sockets can be assigned to one of the AF_UNIX, AF_INET or AF_INET6 address families. The Sockets interfaces are defined in a special library.

Linked into a POSIX application, the connection to the network level is established from these interfaces via the subsystem POSIX and the transport system BCAM.
The SOCKETS system part is integrated in the POSIX subsystem as an own "virtual" file system. The communication tasks that come from the POSIX application are carried out by this system part.
The process via the STREAMS level that is otherwise normal in UNIX has been replaced by a direct connection to the BS2000/OSD transport system BCAM. This results in very short and high-performance paths, with which a high data throughput is achieved. SOCKETS is a flexible and high-performance method of creating network applications.
The SOCKETS interfaces also support the addressing mode IPv6, which among other things allows greater flexibility (addressing width) when it comes to addressing in networks.
Furthermore, the allocation of several IPv4 and/or IPv6 addresses for a host is fully supported (multihoming).



Transport Layer Interface (TLI)

TLI-based network interfaces are available as an option. These interfaces too enable internet access via TCP/IP and UDP/IP.

Like the Sockets interfaces, the TLI interfaces consist of a series of library functions that are linked to the BCAM transport system in BS2000/OSD via a special file system in the POSIX subsystem.

The POSIX file system

As a hierarchically structured file system, the POSIX file system is an extension to the BS2000/OSD. As in the UNIX systems, the user has the option of creating, saving and accessing a hierarchical file system. There are no restrictions in the number of directory levels nor in the number of file directories and files on one level. A POSIX file system can be very well structured and organized, as is the case in UNIX systems. It is possible with the support of 64-bit file systems to have POSIX file systems of up to 1024 GB. The maximum size of a POSIX file is solely restricted by the size of the respective POSIX file system. The program interfaces for the processing of POSIX files are also available in a 64-bit version (e.g. in addition to the open() interface, there is also an open64() interface).

Vorteile des POSIX-Dateisystems

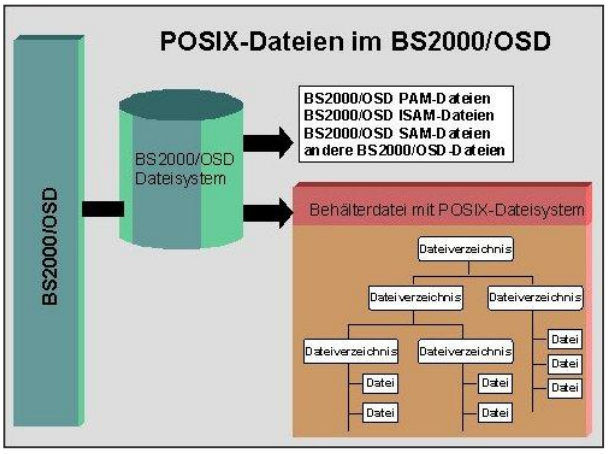
- Hierarchische Struktur des Dateiverzeichnisses
- Leichtes Übertragen von Dateien von einem aktuellen Dateiverzeichnis in ein anderes Verzeichnis.
- Dateien können in einem oder mehreren Dateiverzeichnissen abgelegt werden.
- Im Dateisystem können mehrere Dateien gleichen Namens (in verschiedenen Dateiverzeichnissen) abgelegt werden.

POSIX file systems in BS2000/OSD

POSIX file systems are stored in so-called container files in BS2000/OSD; this is equivalent to the normal storing of files systems in partitions in UNIX systems. The container files are BS2000 PAM files that are located on a pubset. POSIX container files and BS2000/OSD files may be on the same pubset. The container files can be created both by the POSIX administrator and by the BS2000/OSD system administrator. The size of the container file as well as that of the POSIX file systems are defined here. There is a daemon (command fsmond) for the important file systems / (root) and /var, which issues warning messages if the (adjustable) configuration density of these file systems is exceeded.

File systems can be enlarged both online (subsystem POSIX is started) and offline (subsystem POSIX is not started).

The online expansion is offered both as a command (fsexpand) and as part of the administration of POSIX file systems in the POSIX installation program; the offline expansion is a menu item that is also for initial installation in the mask of the POSIX installation.



Journaling for file systems

To enable a fast restart after a system crash POSIX offers the option of keeping a journal with modified meta data (file system journaling).
If such a journal is kept, the modified meta data is then either written to disk in its final position during the restart or rejected, depending on the status at the time of the system crash.
Consequently, the recovery of a consistent state of the file system is accelerated, because according to the journal only the open actions still have to be processed.
The data is available again in the shortest time possible after the crash (a matter of seconds) – fast recovery of data consistency means higher data availability and ultimately also higher productivity for all users and administrators. On the other hand, a conventional file system check would have to search through.

Exchanging files

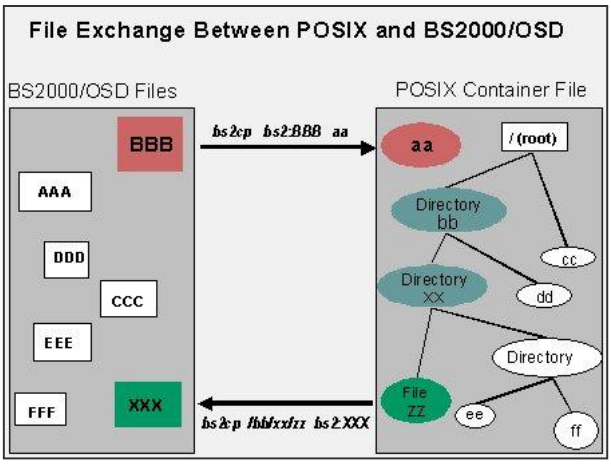
POSIX files are byte-stream-oriented. BS2000/OSD files are record-oriented and/or PAM-block-oriented. POSIX works as a BS2000/OSD subsystem with the EBCDIC format to code texts. However, texts in UNIX or Windows files are coded in ASCII format. ASCII files saved in the POSIX file system can only be sensibly processed by POSIX if they have been converted beforehand. Comprehensive copy and conversion routines are available to enable files of both formats to be used alternately.

The following is used as a means of copying between the BS2000 and the POSIX world: the command bs2cp, with which BS2000 files (SAM, ISAM and PAM) and elements can be copied from PLAM libraries to POSIX and vice versa, or the file system type bs2fs (see section "The BS2000 File System bs2fs").

The *bs2cp* command allows wildcards and can therefore copy multiple files with a single command. Users also have the option of using self-defined code translation tables.

BS2000 provides the SDF command /COPY-POSIX-FILE, which has the same functionality as *bs2cp* but can be called from BS2000. It is consequently not necessary to start the POSIX shell explicitly in BS2000 procedures.

The code set on which the POSIX files are based (EBCDIC or ASCII) is taken into account here, but can optionally be ignored if binary copies are needed for non-text files



The BS2000 file system bs2fs

The BS2000 file system bs2fs allows to access BS2000 files transparently from POSIX and via existing POSIX interfaces (commands and program interfaces). For this purpose a new file system type bs2fs was introduced in addition to the existing file system types ufs or NFS. In order to work with bs2fs, a subset of a user's files is mounted with a bs2fs mount command to a position within the POSIX file system. This mount process makes the BS2000 files accessible to the user. Now he can work on these files with POSIX means, e.g. perform an open() on such a file or apply commands on such files.
In order to access BS2000 files with POSIX interfaces, the files are copied on demand - and not visible for the user - in a container file system especially foreseen for bs2fs (of file system type ufs). This copying task is performed by one or several daemons. A daemon opens this file in BS2000 and copies it to POSIX, at the same time the BS2000 file is locked for other users in BS2000, while the POSIX copy in the container is not locked for other POSIX users. Then operations can be performed on the file, e.g. reads, positions and writes. When the file is closed, it is copied back to BS2000.

Usage scenarios are:
- Browse of BS2000 files resp. PLAM library elements with respect to certain patterns with the mighty POSIX *grep* command.
- Use of the *make* function for efficient generation of programs or program systems.
- Nested procedures (Call of the POSIX shell from the BS2000 command mode, execution of POSIX commands and then return to the BS2000 command mode) can be replaced by pure POSIX shell scripts. File manipulations in BS2000 are replaced by manipulation of these files from within POSIX, once having mounted these files via bs2fs. The amount of switches from BS2000 ito the shell and vice versa is declining.
- Offer BS2000 files in web by simple means.
  The BS2000 file system bs2fs is available as of POSIX version A41.
  NFS can be connected to POSIX as of POSIX version A43.

POSIX takes advantage of BS2000/OSD's excellent security

POSIX is embedded into BS2000/OSD in such a way as to make sure that the security of the overall system is assured at all times. POSIX takes full advantage of the proven security mechanisms in BS2000/OSD. The various elements of the POSIX security concept are detailed below.

User data administration

The POSIX standard defines interfaces for user security controls. These interfaces query specific information about a user before allowing him or her to use an operating system. The following data is used for authentication:

- User ID/login name
- Password
- User number
- Group number
- Initial home directory
- Program to be started

Additional data can be added to these items if necessary. The POSIX user data is stored and administered by BS2000/OSD user administration and is integrated into the BS2000/OSD user data as POSIX user attributes. POSIX user data is managed via the user and system administrator commands of BS2000/OSD.

Group administration

Group administration in POSIX is similar to that in UNIX systems, but differs in some respects from group administration in BS2000/OSD. POSIX and BS2000/OSD groups are therefore maintained in parallel and administered separately. The POSIX groups are administered at the shell level and the BS2000/OSD groups at the BS2000/OSD level.

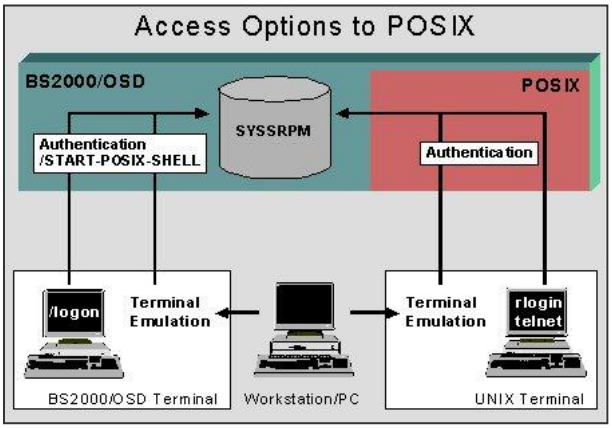
Access protection for container files

POSIX file systems are stored in containers. Containers are BS2000/OSD PAM files in non-key format. They are protected against unauthorized access by the BS2000/OSD standard access control mechanism (USER-ACCESS attributes).

Access protection for files and directories

The following protection mechanisms prevent a user from illegally reading or modifying another user's files and directories in POSIX:

- Each POSIX user has his or her own user ID which can be protected against unauthorized access with a password.
- Users can be combined into groups so that all members of a group have access to the relevant files and directories.
- Files are protected with protection bits against unauthorized reading, writing and execution.



Protection on access from a remote computer

POSIX can also be used from remote computers. Users logging into POSIX from the rlogin command are entered just like local users into the central computer's BS2000/OSD user administration. The BS2000/OSD component SRPM checks the access authorization during processing of the *rlogin*.

Secure shell (OpenSSH) in the interNet Services product

The secure shell (SSH) in POSIX permits secure communication across insecure networks and can replace the insecure r-commands (remote shell), telnet and ftp, which send the entire communication transaction – including the passwords – without encryption.

POSIX security together with SECOS

SECOS satisfies the most stringent system security requirements. SECOS implements access protection with differentiated access classes for a precise access control to the POSIX environment. Access via the multi-computer POSIX commands *rsh* and *rcopy*, for example, can be administered independently of the POSIX rlogin. Access to POSIX via *rlogin* can be set up with SECOS such that only certain stations can be used via *rlogin*.

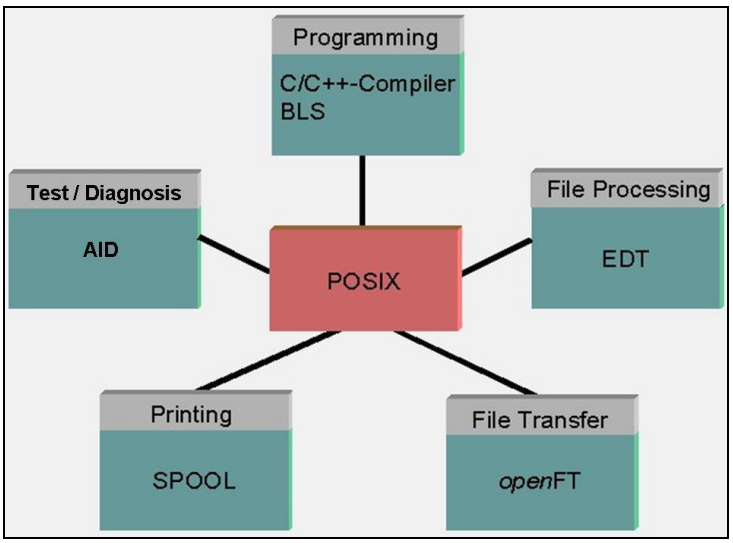
The POSIX commands *at*, *batch*, and *crontab*, which are used to control batch processes, can be restricted by SECOS such that they can be used only by certain users. Events in the POSIX environment can be logged using SAT. This feature makes it possible to log and evaluate more than 40 different types of event including file access activities, process generation and process and privilege changes. The SAT support settings for POSIX events and alarms can be edited selectively to activate or deactivate logging of certain POSIX events in order to reduce the volume of data generated and facilitate more pertinent reports.
SAT logging also covers the 64-bit interfaces.

Dynamic setting of POSIX system parameters

Es ist möglich, Systemparameter von POSIX zu modifizieren ohne POSIX herunterzufahren und die modifizierten Systemparametern für folgende Subsystemläufe zu sichern.
Die POSIX-Systemparameter sind in einem Kernelheader definiert. Sie werden beim Hochfahren von POSIX in die Strukturen des Kernels übernommen und stehen damit POSIX beim Ablauf zur Verfügung. Während des Starts des Subsystems wird die POSIX-Parameter-Datei (SYSSSI-Datei) ausgewertet, die neue Werte der Systemparameter enthalten kann. Zusätzlich können die aktuellen Werte von ausgewählten Systemparametern (z.B. max. Anzahl Terminals für bestimmte POSIX-Zugangsarten, max. Anzahl POSIX-Prozesse/Prozesse pro User, usw.) über ein privilegiertes Kommando verändert werden.
Der veränderte Wert kann optional auch in die SYSSSI-Datei eingetragen und damit automatisch beim nächsten Hochfahren benutzt werden.

Support of POSIX by the system software

A large number of system software products in BS2000/OSD support the POSIX functionality and make useful services available to UNIX applications.



C-/C++ compiler

The sources to be compiled by the BS2000/OSD-C-/C++ compiler may contain POSIX interfaces. All input/output of the compiler in a compilation, preprocessor or syntax-analysis run can be via POSIX files. Mixed cases are also possible. Both ASCII and EBCDIC-coded sources can be compiled. Consequently, sources which were put into BS2000/OSD with NFS can also be compiled. The C/C++ compiler can also be called up in the shell.

BLS (Binder Loader System)

BLS supports read and write access to BLS objects in POSIX and can also be used to load POSIX programs from the POSIX file system. Link and load modules are generated and saved in POSIX files using the binder.

AID

AID is used to debug tasks created with *fork ()*. AID can also debug programs loaded with *exec ()*. Special operands of the *%AID* command enable the user to stop a parent-child process or new program directly after creation or loading and then test it step-by-step with AID commands.

EDT

EDT can open, read, write and close files from the POSIX file system and is able to process files in ASCII code provided an operand is also supplied. The processed data can then be saved in the POSIX file in ASCII code.

SPOOL

SPOOL can also be used to output POSIX files, so every POSIX program can take advantage of the enterprise-wide high-performance printing services of BS2000/OSD from either BS2000/OSD or the POSIX shell. SPOOL supports a broad spectrum of output devices for the SPOOLOUT including line and LED printers and tape devices.

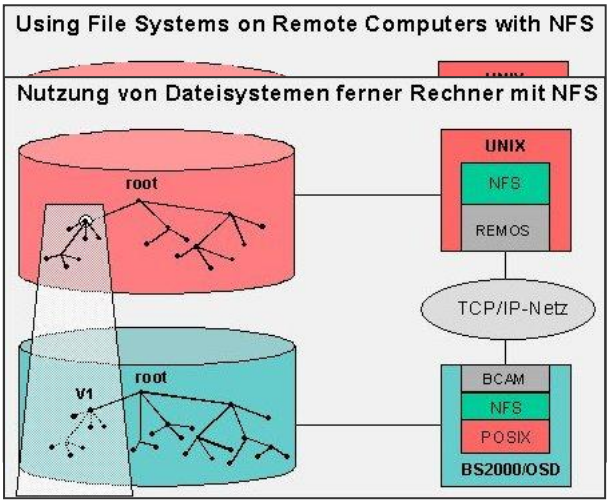
openFT

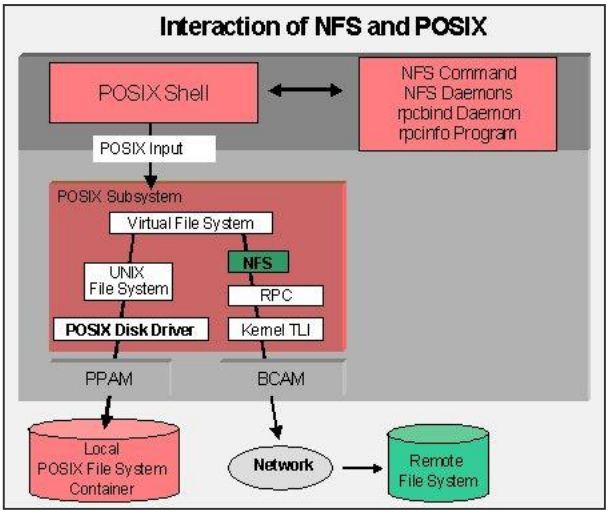
openFT can process and transfer files with different characteristics according file type and operating system. openFT allows to transfer all kinds of BS2000/OSD files (SAM, UPAM, ISAM, POSIX files,..) and PLAM library elements.

Network File System NFS

NFS enables the user to make local files and directories available for processing on a remote computer and to process remote files and directories on his or her own computer as if they were local files. The use of NFS is made possible by the POSIX file system. The file system required is exported from the remote computer using the *share* command and then mounted on the local computer's file system using the *mount* command.

In this way remote computers can use the memory capacities and data backup mechanisms of the BS2000/OSD servers.





NFS components

NFS consists of commands and daemons and uses several administration files.

Commands

NFS adds extra options and specific functions to complement the *mount*, *unmount*, *umount* and *umountall* commands in the POSIX-BC.

Daemons

NFS daemons are system processes that run in the background and coordinate network processes, such as I/O activities for supporting PCs. NFS daemons are started automatically when NFS is started.

Administration files

The administration files support the management of resources and contain either information for the user that is output with the help of commands or information for commands entered into the administration files by the user or by other commands.

NFS V3.0 functions

NFS protocol version 3

Protocol version 2 used in the previous version of NFS permitted only synchronous write mode. NFS 3.0, however, includes protocol version 3, which provides safe asynchronous write mode for much more powerful processing without any reduction in availability.

Network Lock Manager (NLM)

NLM synchronizes access to shared files. The separate NLM protocol can be used to enable, disable and query locks of files and file ranges.

Status Monitor

The Status Monitor is the central collection point for network status information. The Status Monitor makes the network less susceptible to problems and prevents inconsistencies on the various computers within a system.

Support for file systems larger than 2 Gbytes

NFS 3.0 supports 64-bit systems, which means that even file systems containing files larger than 2 Gbytes can be mounted. In BS2000/OSD, NFS V3.0 also supports local large file systems as server.

NFS connection of the bs2fs

The BS2000 file system bs2fs (available since POSIX A41) permits access to BS2000 files from POSIX and with existing POSIX interfaces (commands and program interfaces).
Access to BS2000 data (from open systems) today requires an upstream download or a downstream upload of the bs2fs file (e.g. per file transfer or ftp).
The following extensions are implemented in POSIX A43:

- NFS release of bs2fs file systems
  The bs2fs system permits access to BS2000 files and PLAM library elements from any of the NFS clients. This makes the administration of the replicates unnecessary and provides improved name and location transparency.
- Guaranteeing BS2000 access protection (no NFS-compatible implicit access rights, only those rights which are explicitly issued by the administrator)

The implementation can take place there, where today the cycle "*download from BS2000 – local modification – upload"* is required by means of a file transfer. In addition to improved access times and network load relief (the higher, the lower the modification in comparison to file size), the administration of the resulting replicates can also be omitted. In addition to the function that has been available in the "openFT Explorer" (since openFT V10.0) permitting transparent access to BS2000 files with standard applications (editors, Viewer, ...), the BS2000 files can be processed by any client applications.
The necessary extensions in the shell command *share* in NFS/BS2000 are released as a correction version for NFS V3.0.

Ported products and applications

The use of ported applications like e-mail, Apache or DNS is made possible by the Sockets interfaces that POSIX provides (see p. 5). POSIX therefore occupies a central role when BS2000/OSD is used as an internet server. The most important products and applications for the BS2000/OSD internet server environment are described below.

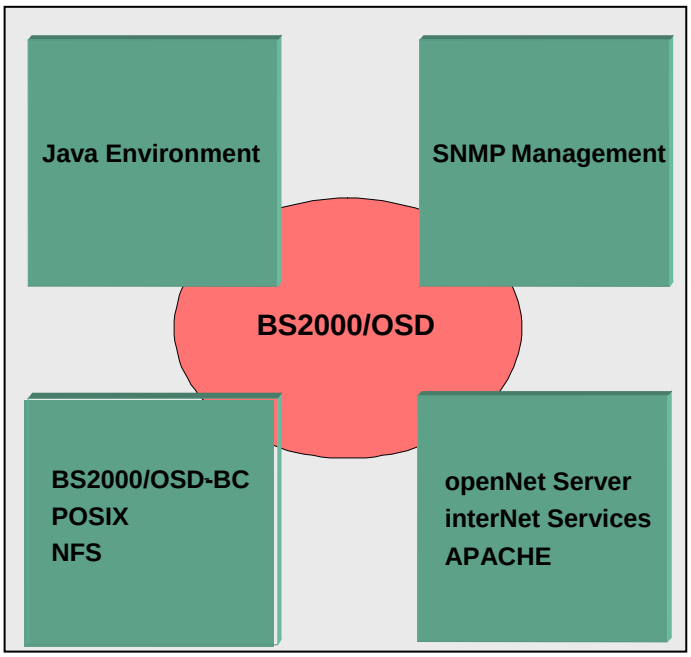
BS2000/OSD as an internet server

Internet opens up new sales avenues for businesses and automates the sales process from ordering to delivery. Such internet-based applications require a powerful and reliable platform. BS2000/OSD is particularly strong in these areas and consequently makes an excellent e-commerce solution.

The Open Group defined the properties and functions required for effective operation of a fully-fledged internet server in a product standard (X98PS – Product Standard Internet Server). BS2000/OSD satisfied these requirements as early as 1999 and was the world's first server to be awarded The Open Group brand for internet servers.

Fujitsu provides products designed especially to turn the high-performance servers of the BS2000/OSD series into web servers. These products are all based around openNet Server, which provides the transport services. The various services of the internet server are implemented by interNet Services (general services and e-mail services) and APACHE (HTTP server). The APACHE product enables protected communication using the SSL protocol.
The Java Environment for BS2000/OSD provides platform-independent execution of applications. The Java Enterprise Edition features additional APIs that are particularly important for the internet.

BS2000/OSD thus supports all of the relevant protocols and internet technologies needed for a powerful internet or intranet server.



Internet services

Fujitsu offers a series of internet services using the TCP and UDP transport protocols on the basis of the openNet Server product. The Open Group defines the internet services of relevance for an internet server in its Product Standard X98PS. The following table shows the main BS2000/OSD internet server services and indicates which service is implemented in which product.

Service Group	Service		Product
TCP/IP communication services	TCP UDP IP, IPV6 ICMP, IGMP	Transmission Control Protocol User Datagram Protocol Internet Protocol	openNet Server
Mail services	SMTP POP IMAP	Simple Mail Transfer Protocol Post Office Protocol Internet Message Access Protocol	interNet Services
File transfer service	FTP	File Transfer Protocol	interNet Services
Name service	DNS,DDNS	Domain Name Service Dynamic DNS	interNet Services
Hypertext services	HTTP, HTTPS	HyperText Transfer Protocol	APACHE
Time service	NTP	Network Time Protocol	interNet Services
Network management	SNMP	Simple Network Management Protocol	SNMP Basic Agent BS2000
Terminal service	TELNET	Terminal emulation	interNet Services und POSIX-PC
Security Service Open SSH	SSH	Secure Shell	Internet Services

openNet Server

openNet Server is the central communication manager in BS2000/OSD. It provides communication and transport protocols in connection with interoperability that are particularly important for a client/server-oriented application architecture. openNet Server features communication services for all relevant networks. It implements a shared transport service that enables communication with the TCP/IP and ISO product standards as well as with the specific NEA protocols of Fujitsu.

Security in openNetworking

Extensive protection is possible with the functionalities offered in openNetworking. Access to a number of services (TELNET, FTP, Mail, http and DNS) as well as the transfer of data between server and client can be protected using symmetric or asymmetric encryption algorithms. Data encryption and the authentication of communication partners for socket applications (such as FTP, TELNET und APACHE) is also enabled through the porting of OpenSSL to BS2000/OSD. The encryption itself can be done by OpenSSL or by the "openCRYPT™" products.

The Internet standard services

The products in the interNet Services package are most ports of corresponding standard internet products from the open UNIX-based system world that have been adapted to the specific requirements of BS2000/OSD. This ensures not only a consistent user interface, but also administrability and interoperability across system boundaries. The interNet Services package includes the basic functions, services and protocols necessary for operation on the internet. The components of interNet Services are continuously enhanced and adapted. Selected components of the interNet Services product are listed below.

FTP (File Transfer Protocol)

FTP enables the transfer of all types of files. BS2000/OSD can provide both server and client functionalities using FTP. The following features are included along with the standard protocol:

- Support for BS2000/OSD file formats (SAM, PAM)
- Use of code tables for converting from EBCDIC to ASCII and vice versa
- Secure transmission using encryption via SSL (optional)
- Additional security functions by linking to the optional openFT-AC product
- Expanded system exit for individually-programmable code conversion in clients and servers during file transfers
- SNMP connector for the FTP server
- Restart mechanism for file transfers
- Batch support for the FTP client

The FTP client may as an alternative be run under POSIX, in which case the FTP client operates in the local directory of POSIX and the transferred files no longer have to be copied from the BS2000 directory into the POSIX directory if they are to be processed further under POSIX.

DNS (Domain Name Service)

The Domain Name Service is a distributed replicated database with DNS servers and DNS clients (resolvers). Here too BS2000/OSD provides both server and resolver functionalities for its users. As an option, the requests to remote DNS servers can be sent with signature.

OpenSSH

OpenSSH is a secure alternative to the utilities rlogin, rsh and rcp and the programs telnet and ftp.
OpenSSH encrypts the entire network traffic (including passwords) and ensures mutual authentication of the communication partners.

The mail services

Mail server (mail transfer agent)

The sending and receiving of e-mail is undoubtedly one of the most important services on the internet. The mail server is responsible for transferring e-mail messages over the network and distributing them to the mailboxes. The mail service is based on SMTP (Simple Mail Transfer Protocol). The special protocols POP3 (Post Office Protocol 3) and IMAP (Internet Mail Access Protocol) enable access to mailboxes from a remote computer (PC). The electronic mail service in BS2000/OSD is implemented with an SMTP mail server, a POP3 server and an IMAP server.

Mail client (mail user agent)

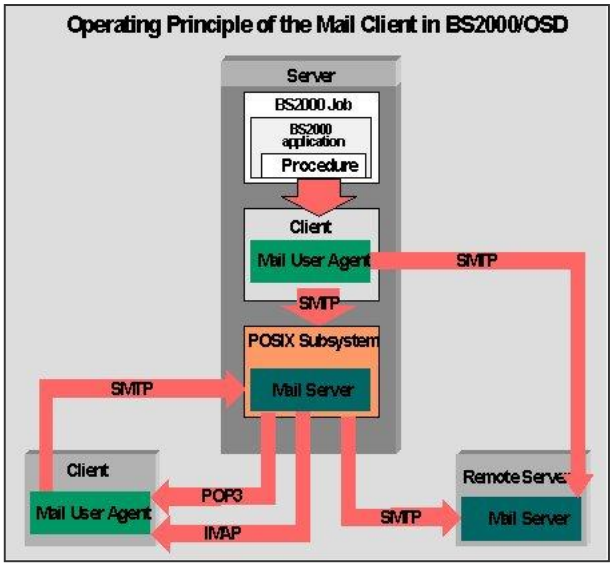
Mail sender

It is also possible to send and process files as e-mail. BS2000/OSD includes a local user agent for this purpose that permits users to send e-mail messages directly from a procedure. Particularly useful applications of this feature include automated list distribution and the automatic issuing of alerts in the event of a fault developing. Alerts can be sent as e-mail messages to the local mail server in POSIX or to a remote mail server.

Mail reader

The mail reader allows users to retrieve and process e-mail messages with the access services (POP3 and IMAP) in BS2000/OSD. BS2000/OSD provides procedure and program interfaces for this purpose. These interfaces both enable the user to access the message header, message body and attachments of an e-mail message.

The mail services in BS2000/OSD are part of the interNet Services package.



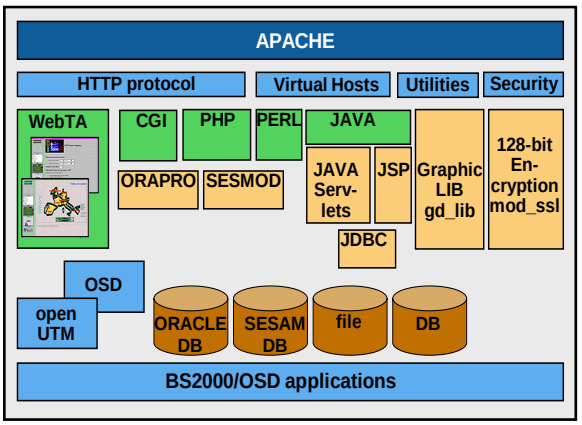
WWW services – the APACHE web server

Apache is the world's most used web server. APACHE (BS2000/OSD) is supplied together with the BS2000/OSD basic configuration. APACHE is in itself structured in such a way that convenient access to information is possible in the form of hypertext and hypermedia links.
APACHE V2.2 can be used to also transfer web pages in encrypted form and can thus also be used for sensitive applications.

The basic characteristics of the APACHE web server are:

- Virtual hosting
  Virtual hosting makes it possible to present different elements of a website's content independently. Virtual hosting with APACHE is optionally based on IP addresses, IP port numbers or host names.
- Access restrictions
  Site access can be restricted at directory/file level by the APACHE server, which accordingly maintains a list of user names and passwords for each virtual directory. Additional protection is provided by granting access to a virtual directory not on the basis of user name, but rather on the requesting client's IP address.
- Persistent connections
  Connections can be maintained for a certain period of time without any need to re-establish them for each document or image.
- Spelling correction
  APACHE corrects simple typing errors and capitalization on request.
- Content negotiation
  The server automatically matches its pages dynamically to the user profile, so pages can be presented to the client in the appropriate language.
- Full Java support
  APACHE V2.2 supports Java servlets and Java server pages (see page 154), which can thus easily be integrated into web presentations.

- SESAM and ORACLE database connection
  A range of new features that enable SESAM and ORACLE files to be processed in BS2000/OSD from within PHP scripts were added to the PHP script language in APACHE. This provides a straightforward way to make database content available on the web and realize write access to the database. The PHP code is executed exclusively on the server, with only the HTTP code being transmitted to the client. The application logic remains hidden from the web user, so there are no security issues involved.
- Perl connection
  The Perl scripting language popular in the UNIX system environment for execution automation is fully integrated into the APACHE web server, so high-performance script execution is assured. Perl can also be installed and used without APACHE and can be employed to automate execution in POSIX.
- WebDAV
  An expansion of HTTP, WebDAV (Web-based Distributed Authoring and Versioning) enables users in different locations to access central documents and process them like local documents. The access rights are distributed using standard APACHE mechanisms.
- Support for IPv6
  openNet Server is used to support the protocol IPv6 in BS2000. Together with APACHE it is also possible to accept connections via IPv6.



Portability with Java

Mainframes are increasingly becoming a powerful e-business platform. Scalability, reliability and security – the main strengths of BS2000/OSD – are major factors in areas where constant availability is essential. Java, “the internet programming language”, plays a key role in this connection. Developed as a “write once, run everywhere” language, Java has spread rapidly and is attractive for use under BS2000/OSD for the following reasons in particular:

- The program code can be created on a local PC and executed on the mainframe.
- Java can be used to develop server applications without any detailed knowledge of BS2000/OSD.

Java benefits from the secure high-availability system environment of BS2000/OSD. The Java source code is translated into a Java byte code, which is identical on all platforms. This byte code is interpreted by a Java Virtual Machine (also referred to as the Java Runtime Environment). The runtime environment sits on top of the processor and operating system level and makes the platform-independent execution of the Java byte code possible. BS2000/OSD received the “Java compatible” logo with the Environment for Java (JENV (BS2000) V1.1), which completed the first step in the creation of a Java infrastructure for business-critical applications under BS2000/OSD, in 1998.

Java 2 Platform Standard Edition (Java SE)

JENV (BS2000/OSD) V6.0 is an implementation of the “Java Platform, Standard Edition” (Java SE) for the BS2000/OSD environment for Java ™ V6.0. It was released in July 2010 and has been included in new deliveries as a delivery component of operating systems as of BS2000/OSD-BC V7.0.
JENV V6.0 is sold, as of OSD V7.0 or the X86 version as of OSD V8.0, as part of the basic configuration of the operating system. This means that JENV is delivered with BS2000/OSD-BC and OSD/XC.

JENV (BS2000/OSD) V6.0 includes a runtime environment (JRE), which meets the relevant specifications:

"The Java Language Specification, Third Edition"
"The Java Virtual Machine Specification, Second Edition" version-specific API Specification "Java 2 Platform Standard Edition 6.0 API Specification,"

The adherence to these specifications was confirmed by Oracle America Inc. by assigning the logo “Java Compatible”.

The components for executing Java programs include:

- Java Interpreter (java, also called Java Virtual Machine JVM) in conjunction with various utilities
- A highly-optimized client version of the HotSpot VM
- The client version of the HotSpot VM is realized in JENV.

A version optimized for this platform has been provided for high-performance operation on SX/SQ systems. The integrated HotSpot Client VM directly generates Sparc-/X86 code.

The product also contains a development environment (JDK) with various development tools. These can be used to develop applications or applets which conform to the above API specification.

The components used for developing Java programs inlcude:
- Java Compiler (javac)
- Java Debugger (jdb)
- Java-Classes

- The program for Java archive files (jar), Appletviewer, and other utilities.

The Java Development Kit V6 is supplied together with the BS2000/OSD basic configuration in a version that has been optimized for the respective hardware. JDK gives the programmer a basic system for developing Java applications. The various components of the JDK are:

- HotSpot Compiler
- Java-Interpreter
- Debugger
- Java class libraries
- Diverse other tools

JENV (BS2000/OSD) V6.0 also includes the package JRIO. This package is a collection of Java classes for direct handling of files that have a record and/or a block structure and which are used for the record or page-oriented input/output to such files. JENV V6.0 supports the files of the BS2000 data management system.

JENV is installed in the BS2000-POSIX file system and is normally used in the POSIX environment (POSIX Shell). It can also be controlled via procedures from within the BS2000 environment.

Java Platform Enterprise Edition (Java EE)

Enterprise-wide applications require access to many different distributed middleware services. The Java Enterprise Edition defines the necessary program interfaces and supports the application programming model (APM), which divides an enterprise solution into three parts.

Components

The JavaBeans, servlets and Java Server Pages basic technologies are available for component-based solutions:

- JavaBeans
  JavaBeans enable users to create server-based solutions independently of the underlying database, transaction system and other application components.
- Servlets
  Servlets run in the server's Java environment. They enable access to all local server resources, such as SESAM or Oracle databases. The result of the application is transmitted as HTML code to the client, which consequently does not need a Java runtime environment. Servlets can be used to execute high-performance web services.
- Java Server Pages (JSP)
  Java Server Pages are used to generate dynamic content. The Java code is embedded directly into the HTML files. Full use of the Java functionality can be made in the server pages.

The support for Servlets and Java Server pages is implemented by porting the Apache Tomcat.

Containers

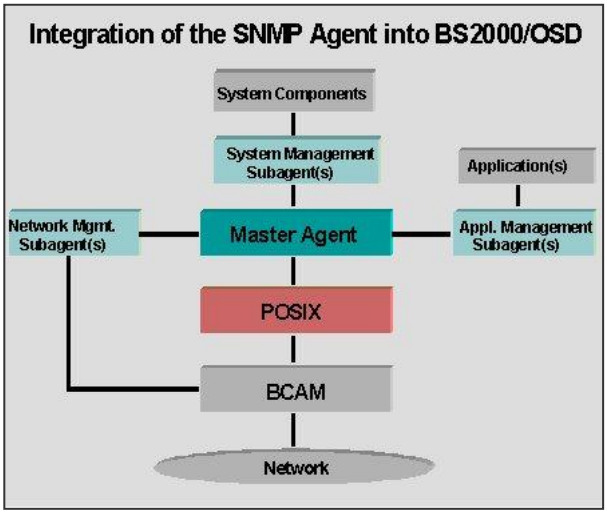
Containers can be used to create very powerful runtime environments for Enterprise JavaBeans, servlets and Java Server Pages. The containers are provided together with a Java EE-compatible application server as for example the Oracle Application Server and the Apache web server.

Connectors

The Java EE APM enables users to construct connectors that make existing legacy solutions usable in a Java EE application. For this purpose, Fujitsu offers the BeanConnect connector family. This means that full use of existing investments can still be made even in new business processes.

SNMP management

SNMP (Simple Network Management Protocol) was originally developed as a protocol for network management services on the TCP/IP internet. SNMP management has proven to be highly efficacious, however, and its application range has therefore increased substantially. Today, SNMP is used universally for all areas from cable management and power supplies to system and application monitoring.
SNMP makes it possible to integrate the normally separate network, system and application management of the individual systems at a data center into a single management environment. This environment serves as a central control station and has a graphical user interface for straightforward and convenient monitoring of the various components to be managed.



Management platform and agents

The heart of an SNMP management environment is the management platform on which the SNMP manager resides. The management platform is an application that communicates with the SNMP agents via SNMP over a TCP/IP network. The agents control the components to be managed and relay information about them to the SNMP manager.

Master/subagent principle

The SNMP agents in BS2000/OSD are based on modern EMANATE technology and are structured according to the master/subagent principle. The master agent is responsible for the central execution of basic tasks, such as processing the SNMP protocol, security functions and monitoring communication.

The actual monitoring is carried out by the subagents, which work directly with the respective components. The subagents are closed systems and do not interfere with each other. The subagents can only operate when the master agent has been started.

The principal benefits of the master/subagent concept are improved availability and enhanced user-friendliness with respect to maintenance and modifications. The EMANATE technology provides powerful development tools that make creating new subagents a perfectly straightforward process.

The master agent in BS2000/OSD has been ported from a UNIX system based implementation on the basis of POSIX and the subagents for BS2000/OSD use the POSIX interfaces.

Integration of the storage systems Fujitsu ETERNUS DX and EMC Symmetrix in BS2000/OSD

Storage Host Component for BS2000/OSD

The storage host component for BS2000/OSD (SHC-OSD) provides information services and commands to control the storage systems Fujitsu ETERNUS DX and EMC Symmetrix:

- SHC-OSD provides commands to control the replication functions for ETERNUS DX storage systems:
  Local replication with clones with EC (Equivalent Copy) provides local continuous mirroring on a volume basis. The synchronous remote mirroring with REC (Remote Equivalent Copy) supports mirroring on a volume basis between 2 or more ETERNUS DX systems. The function local replication with SnapOPC+ provides the option of creating one or more snapshots of the entire volume.
  SHC-OSD also provides information about the ETERNUS DX configuration and about local and remote mirroring.
- SHC-OSD controls the functions SRDF and TimeFinder for EMC Symmetrix:
  The Symmetrix function SRDF (Symmetrix Remote Data Facility) enables the copying of data to another (remote) Symmetrix system. The Symmetrix function TimeFinder allows the creation of additional copies of volumes within a Symmetrix system.
  SHC-OSD supplies selected information about the Symmetrix configuration as well as about the current status of SRDF and TimeFinder processing.

POSIX, POSIX-SOCKETS and PTHREADS must be available in order to operate SHC-OSD. This requirement exists for the following components:

- SHC-OSD requires among other things the SYMAPI components in order to support the different storage systems. SYMAPI consists of the driver components, which are required to support all storage systems and the product SYMAPI, which is ported to BS2000 in the POSIX environment, from EMC2 Corporation for the control and monitoring of Symmetrix systems.

- The software licenses required to use the replication functions of the ETERNUS DX storage system are controlled via the products CM-LR and CM-RR, which are supplied in addition to SHC-OSD for BS2000/OSD and are installed in POSIX.

Storage Control Center Agent for BS2000/OSD

The product SCCA-BS2 (Storage Control Center Agent for BS2000/OSD) implements the integration of the BS2000/OSD and connected Symmetrix systems into the product family EMC Ionix ControlCenter, which offers general functions for the presentation, monitoring, automation and provision of storage resources in a heterogeneous storage and server environment. SCCA-BS2 is also based on the strategic Symmetrix interface SYMAPI and thus on POSIX and PTHREADS.

SCCA-BS2 supplies the information specific to the BS2000/OSD operating system in relation to servers, disk configuration and SAN connection to the central server of the EMC Ionix ControlCenter. SCCA-BS2 thus offers an ideal supplement to SHC-OSD for Symmetrix systems.

Published by department:

Elfi Drayer
Phone: ++49 (0)89 62060 1973
Fax: ++49 (0)89 62060 329 1973
elfi.drayer@ts.fujitsu.com
ts.fujitsu.com

Partner login
partners.ts.fujitsu.com