OS/390 UNIX Technology

SESSION 2921/2922

Kershaw Mehta

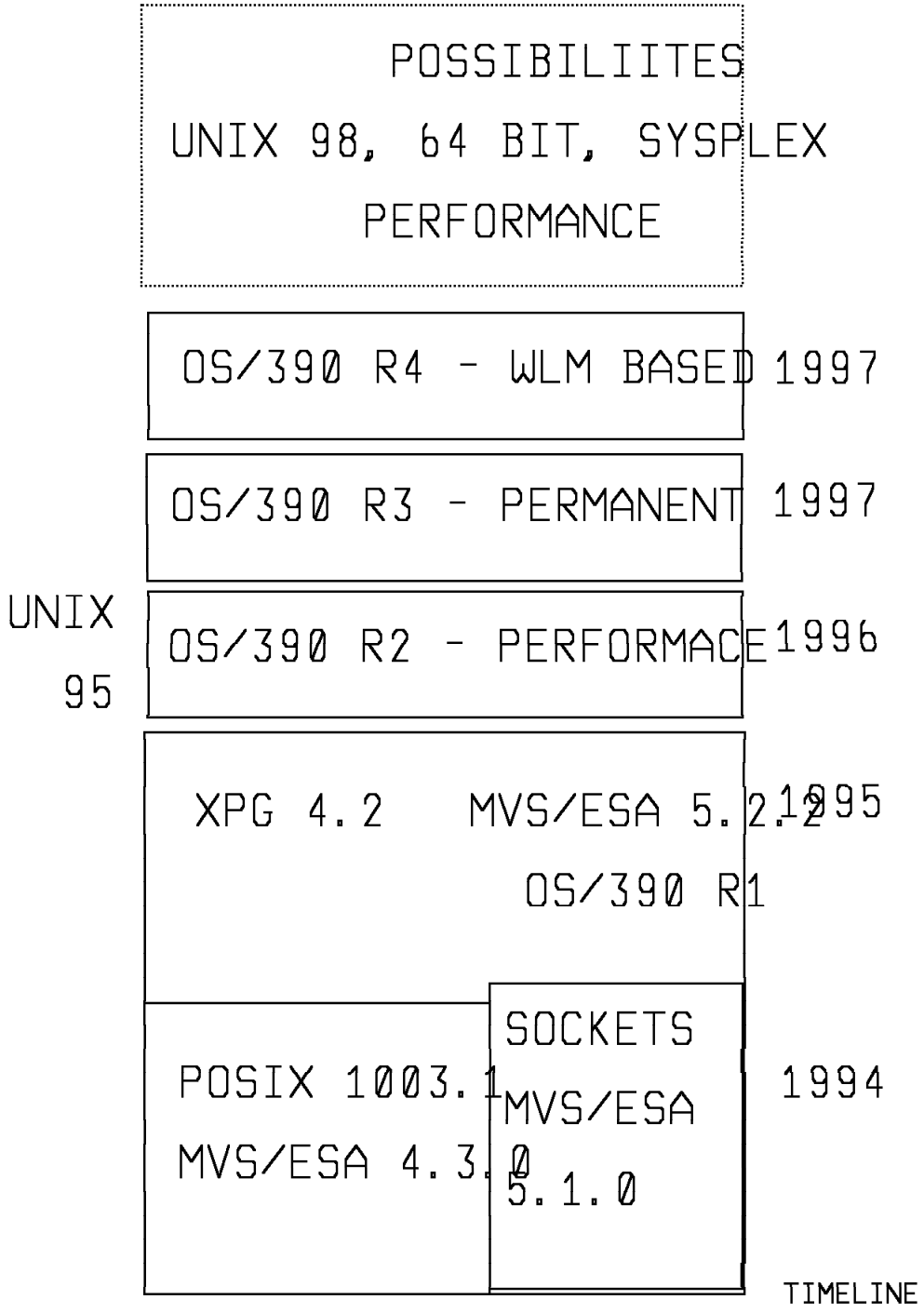
kershaw@us.ibm.com

August 18, 1998

Table of Contents

Table of Contents

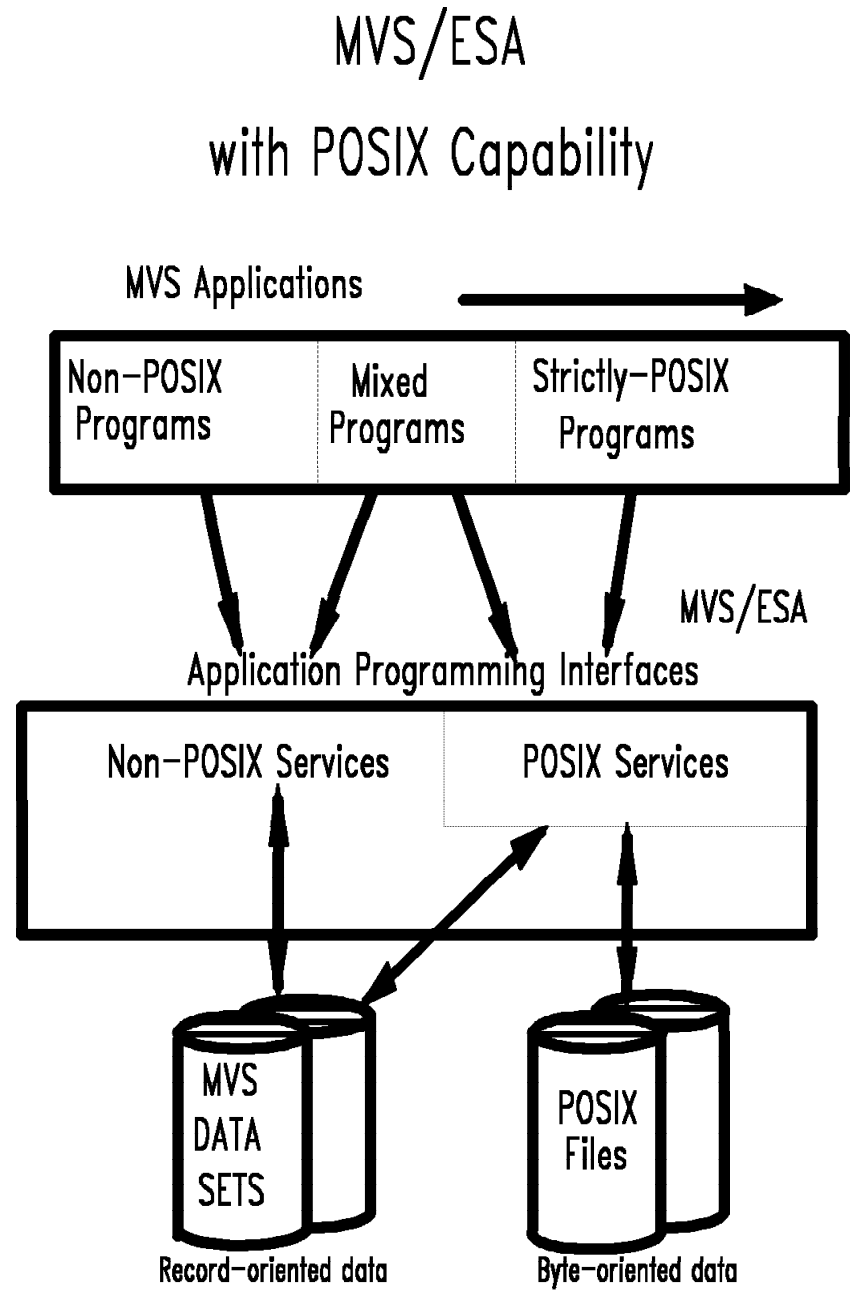
Standards Supported



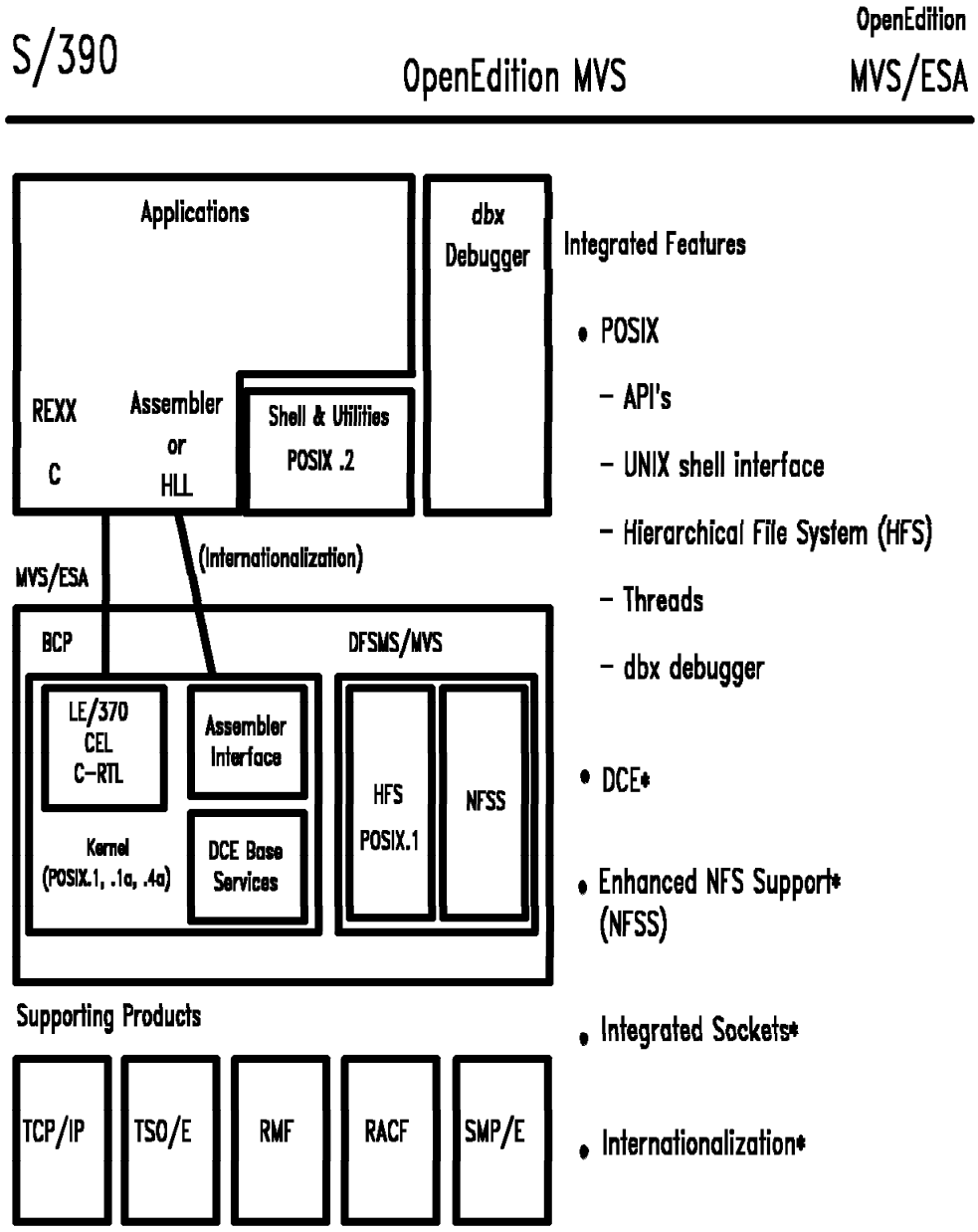
Old and New

- MVS/ESA 4.3 - POSIX 1003.1 fork,exec,signals
- MVS/ESA 5.1 - POSIX 1003.1 sockets, daemon support
- MVS/ESA 5.2.2 - SPEC1170 spawn, shmem, msgq, semaphore, ...
- OS/390 R1 same as MVS/ESA 5.2.2 from OS/390 UNIX perspective
- OS/390 R2 - Performance and WEB requirements
- OS/390 R3 - Performance, WEB, NOTES, SAP requirements - WLM
- OS/390 R4 - APPC elimination
- OS/390 R5 - Performance, TCP/IP Stack, Multiproc/Multiuser

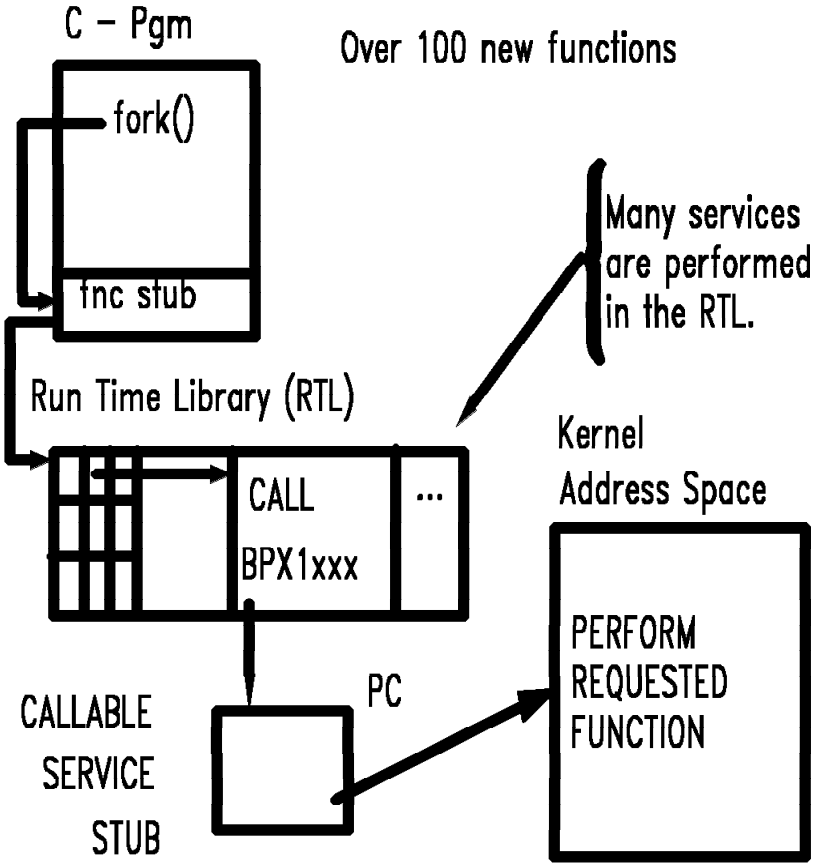
Supported Environments



OpenMVS Product Set



CALLABLE SERVICES



Useful Services for Assembler Programmers:

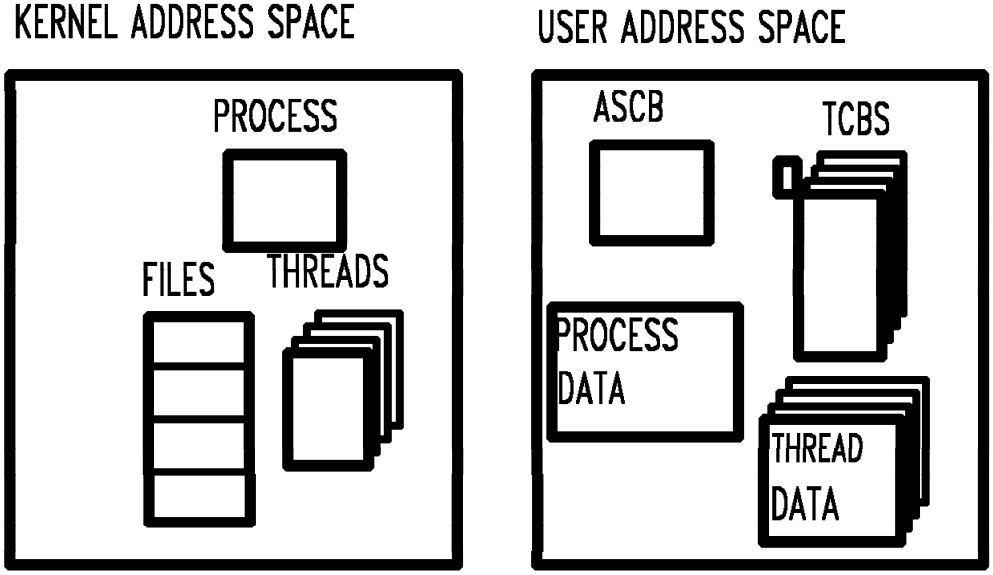
open close read write pipe mkfifo

fork exec waitpid

callsrv2

DUB THEE POSIX

DUB – Build environment to support kernel syscalls

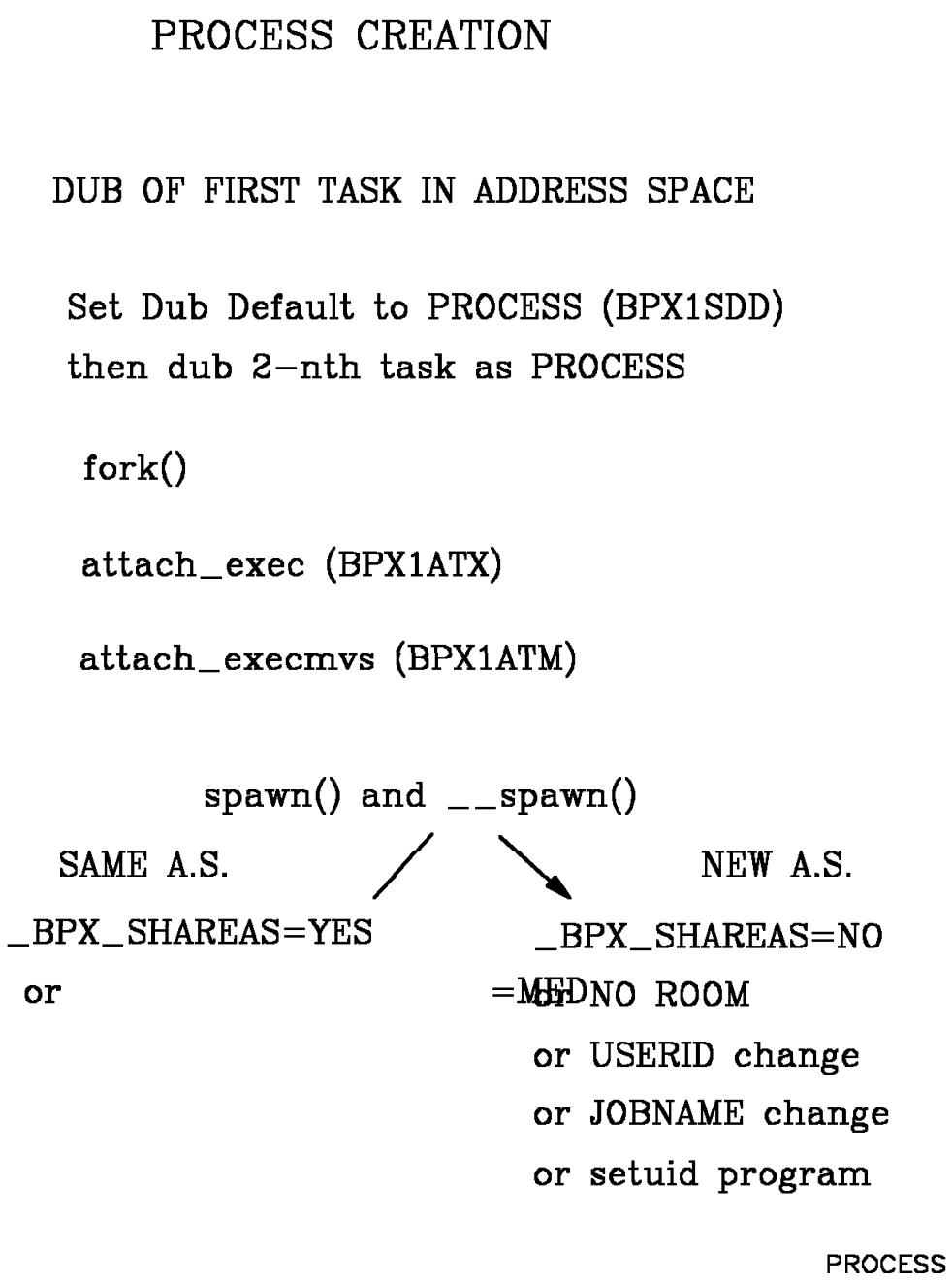


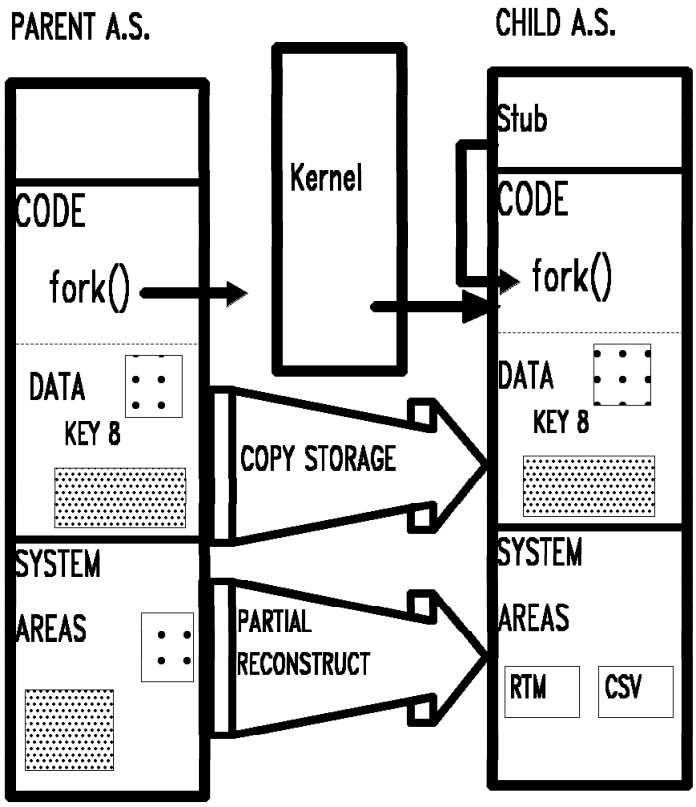
Dub occurs on:

- First kernel call (C main() -> BPX1MSS)
  For each task. Process done on first task.
- In child as a result of fork
- In new task resulting from exec

Undub occurs by call to BPX1MPC, EOT, or EOM.

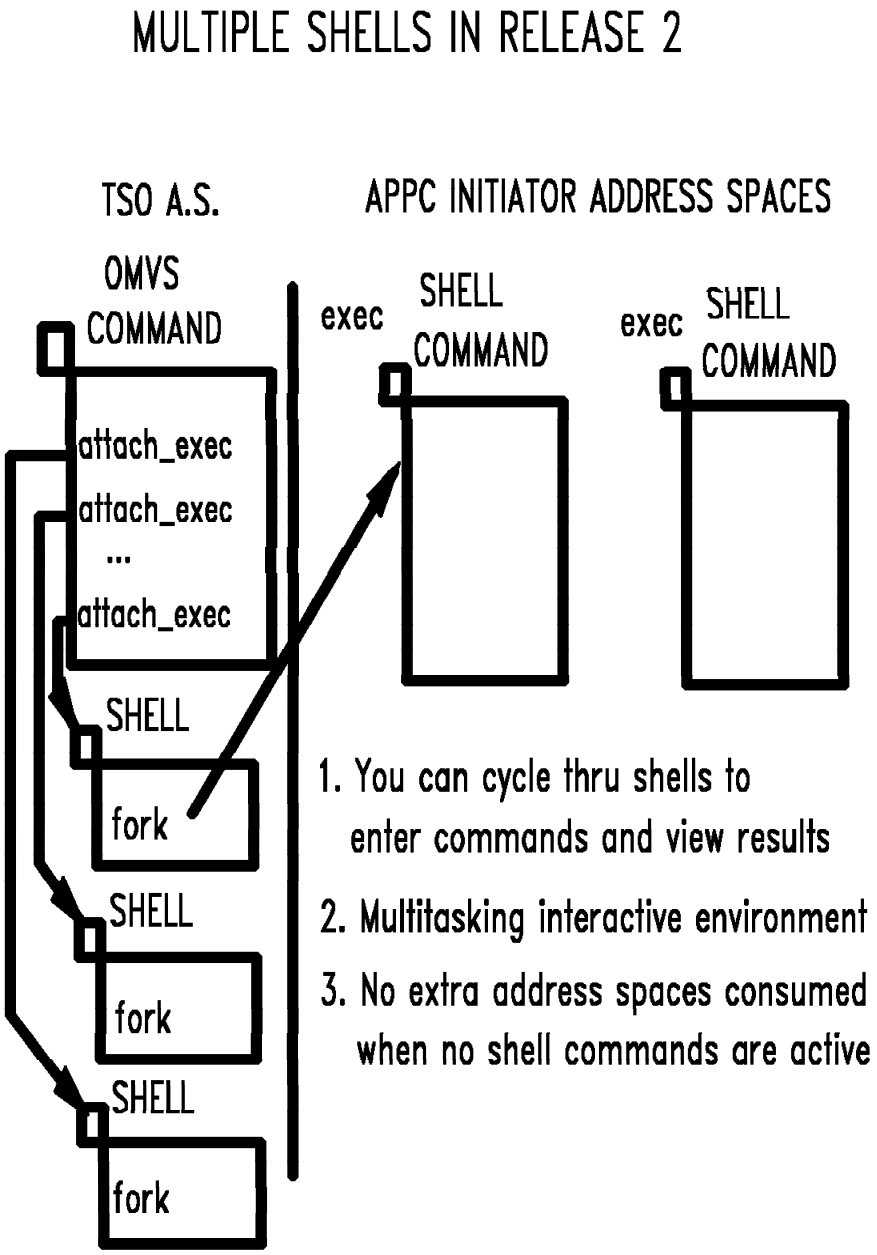
DUB


PROCESS CREATION
DUB OF FIRST TASK IN ADDRESS SPACE
Set Dub Default to PROCESS (BPX1SDD)
then dub 2-nth task as PROCESS
fork()
attach_exec (BPX1ATX)
attach_execmvs (BPX1ATM)
spawn() and __spawn()
SAME A.S.
NEW A.S.
_BPX_SHAREAS=YES
or
_BPX_SHAREAS=NO
=MED
or NO ROOM
or USERID change
or JOBNAME change
or setuid program
PROCESS

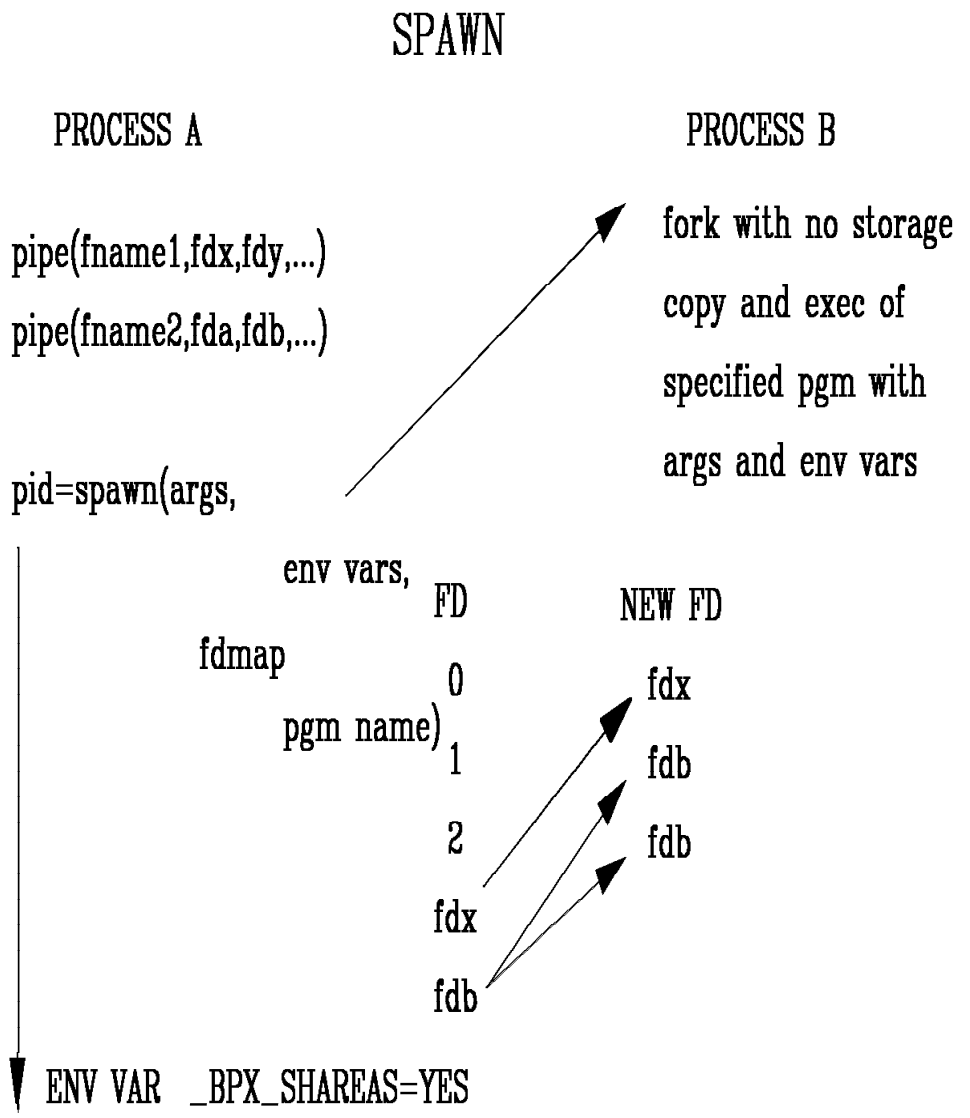


- User subpools and code copied
- User recovery routines and contents supervisor structures propagated – Including STEPLIB
- MVS userid and accounting data propagated

Multiple Processes in Address Space



Spawn

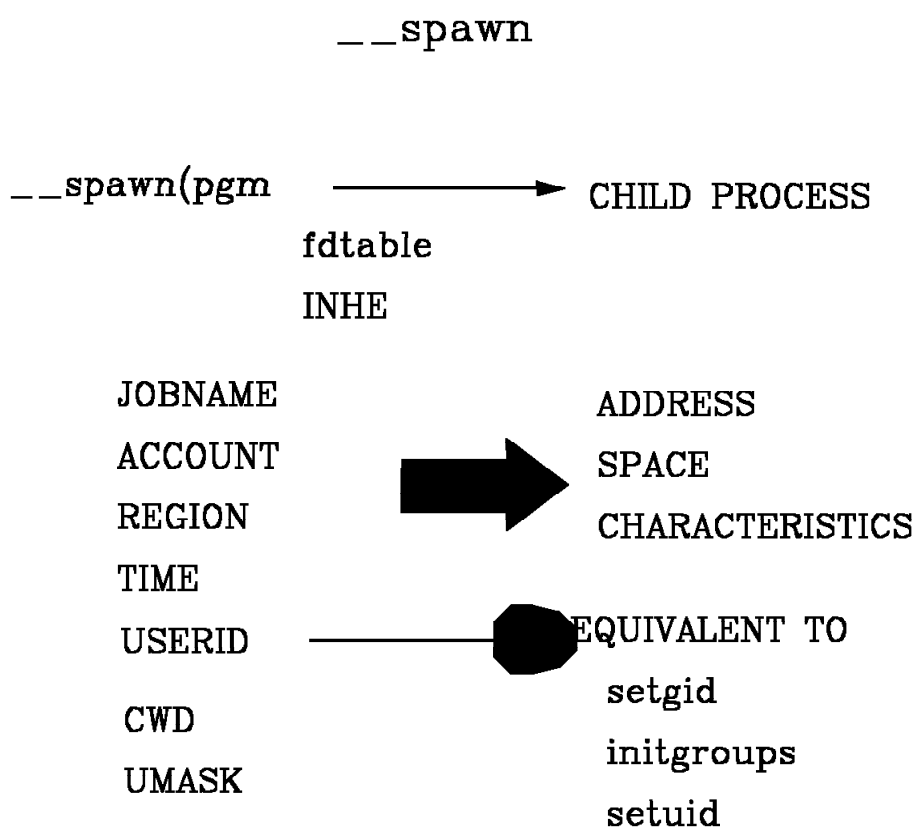


PROCESS B

ATTACH_EXEC for a local process, creates new task and runs pgm with args and env vars.

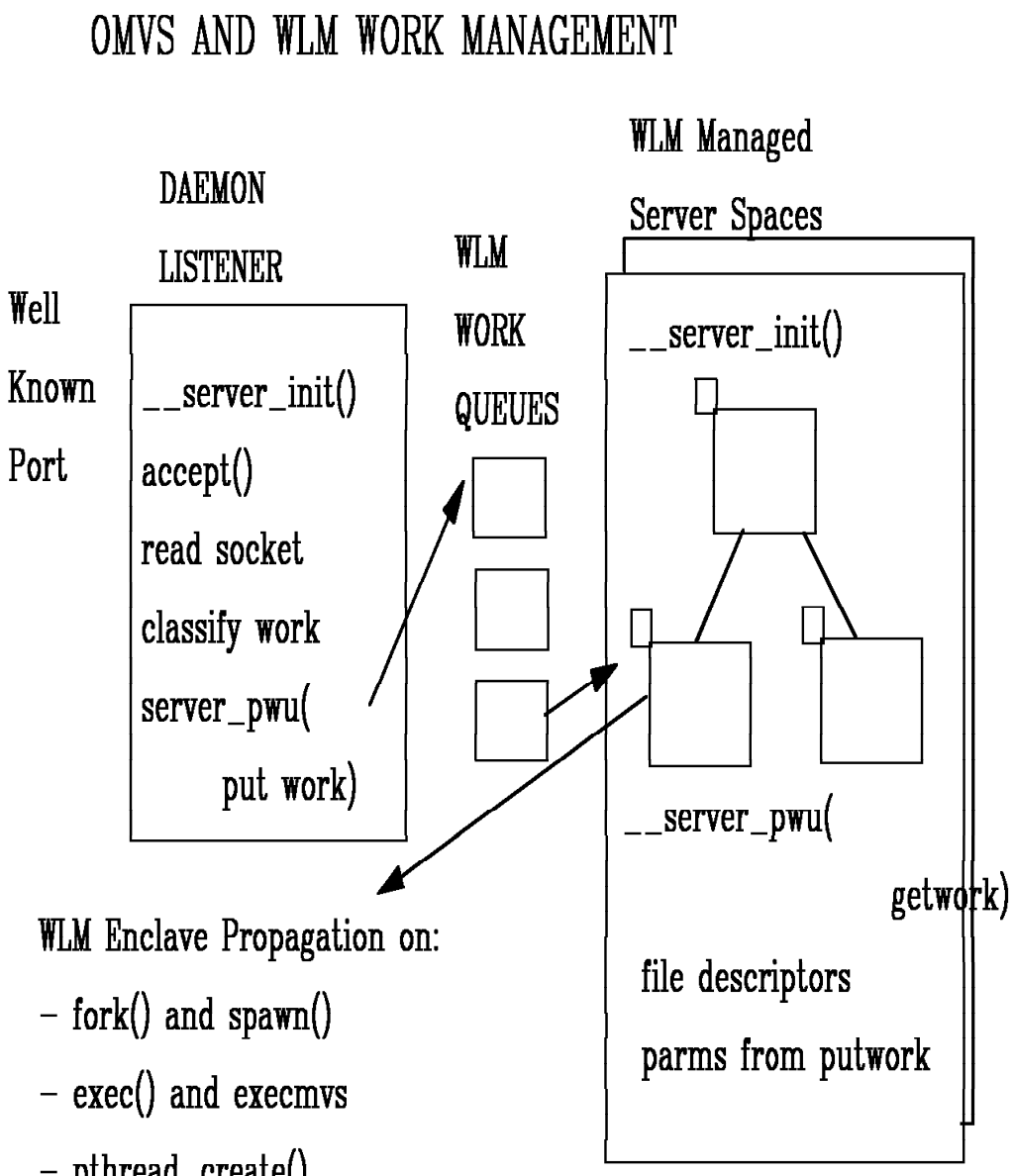
SPAWN

Spawn with Userid (R2)

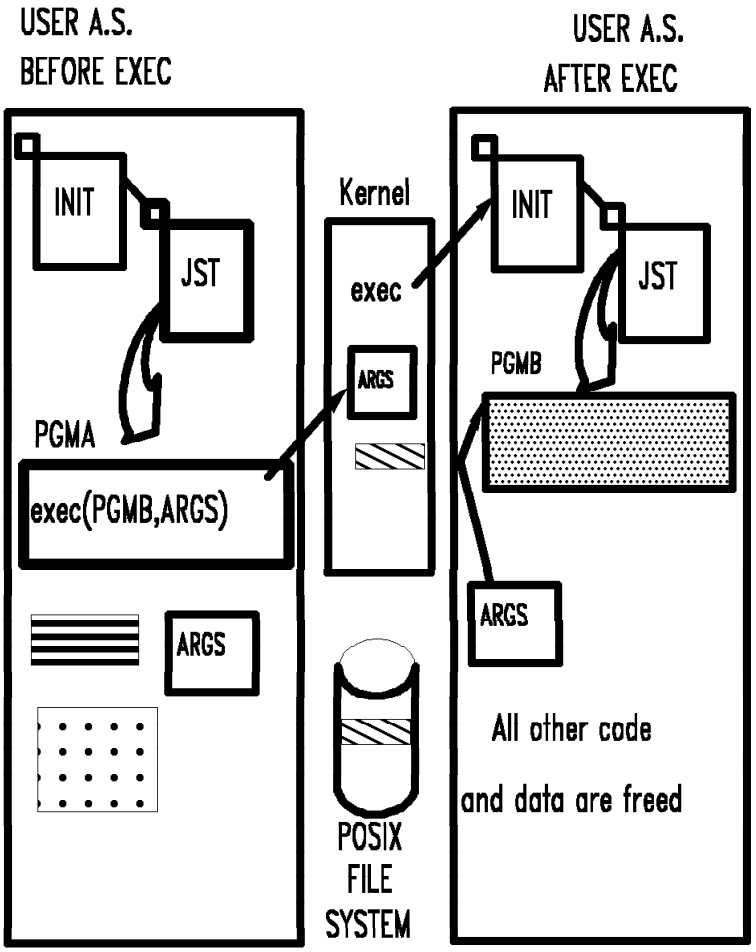


ACCOUNT can be used to give similar server processes different performance behavior by tailoring WLM by userid/account.

SPAWN2


OMVS AND WLM WORK MANAGEMENT
WLM Managed
Server Spaces
DAEMON
LISTENER
WLM
WORK
QUEUES
Well
Known
Port
__server_init()
accept()
read socket
classify work
server_pwu(
put work)
__server_init()
__server_pwu(
getwork)
file descriptors
parms from putwork
WLM Enclave Propagation on:
- fork() and spawn()
- exec() and execmvs
- pthread_create()
- attach_exec, attach_execmvs, local spawn
OMVSWLM1

EXEC

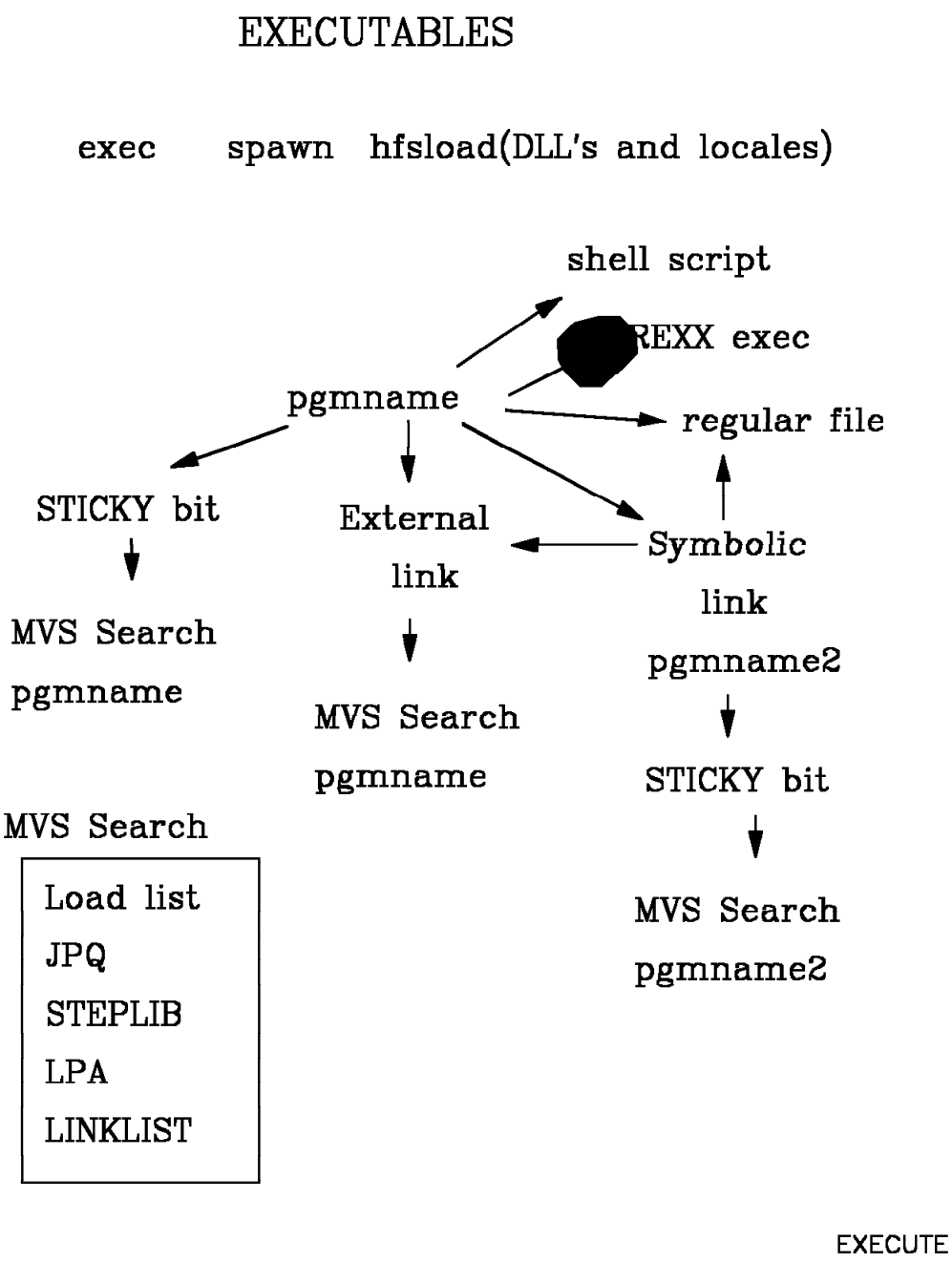


exec terminates all user tasks and creates

a new Job Step Task (JST)

EXECOVR1

exec ...


EXECUTABLES
exec spawn hfsload(DLL's and locales)
shell script
REXX exec
pgmname
regular file
STICKY bit
External
link
Symbolic
link
pgmname2
MVS Search
pgmname
MVS Search
pgmname
STICKY bit
MVS Search
pgmname2
MVS Search
Load list
JPQ
STEPLIB
LPA
LINKLIST
EXECUTE

STEPLIB SELECTION PROCESS

exec(pgm,...)

STEPLIB determination priorities:

1. STEPLIB environment variable -> dynamic STEPLIB
2. Current STEPLIB propagated to new Job Step

   STEPLIB also propagated on fork

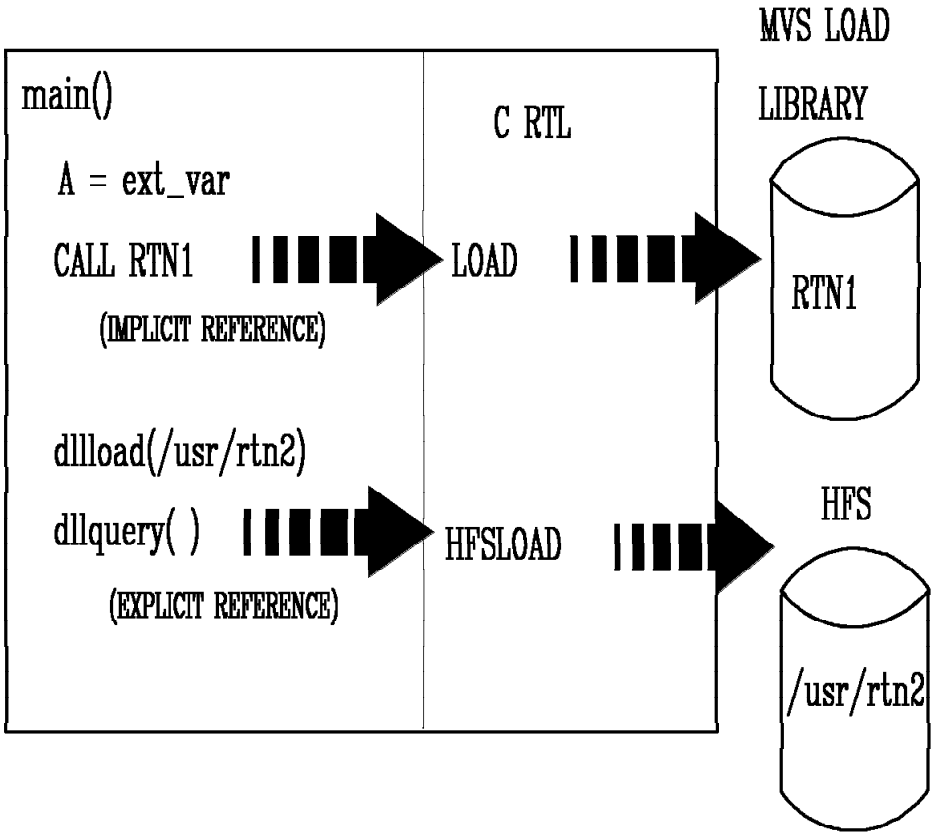
   * Steplib data sets must pass authorization check when a SETUID program is being invoked.

CAPABILITIES and RECOMMENDATIONS

1. Put RTL in LINKLIB for primary users
2. Put frequently called RTL modules in LPA
3. Use STEPLIB environment variable to:
   - test new level of the RTL
   - lock in an old level of RTL for an application

STEPLIB

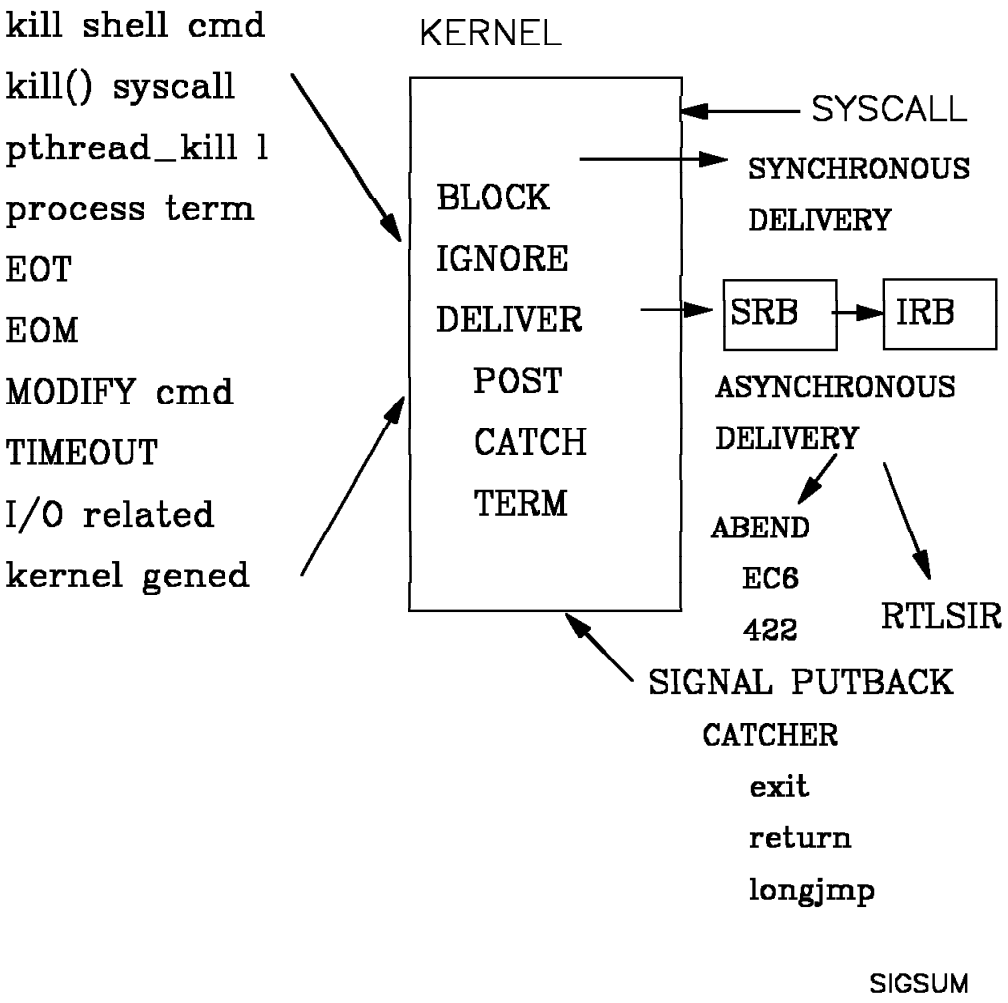
Dynamic Link Library - DLL

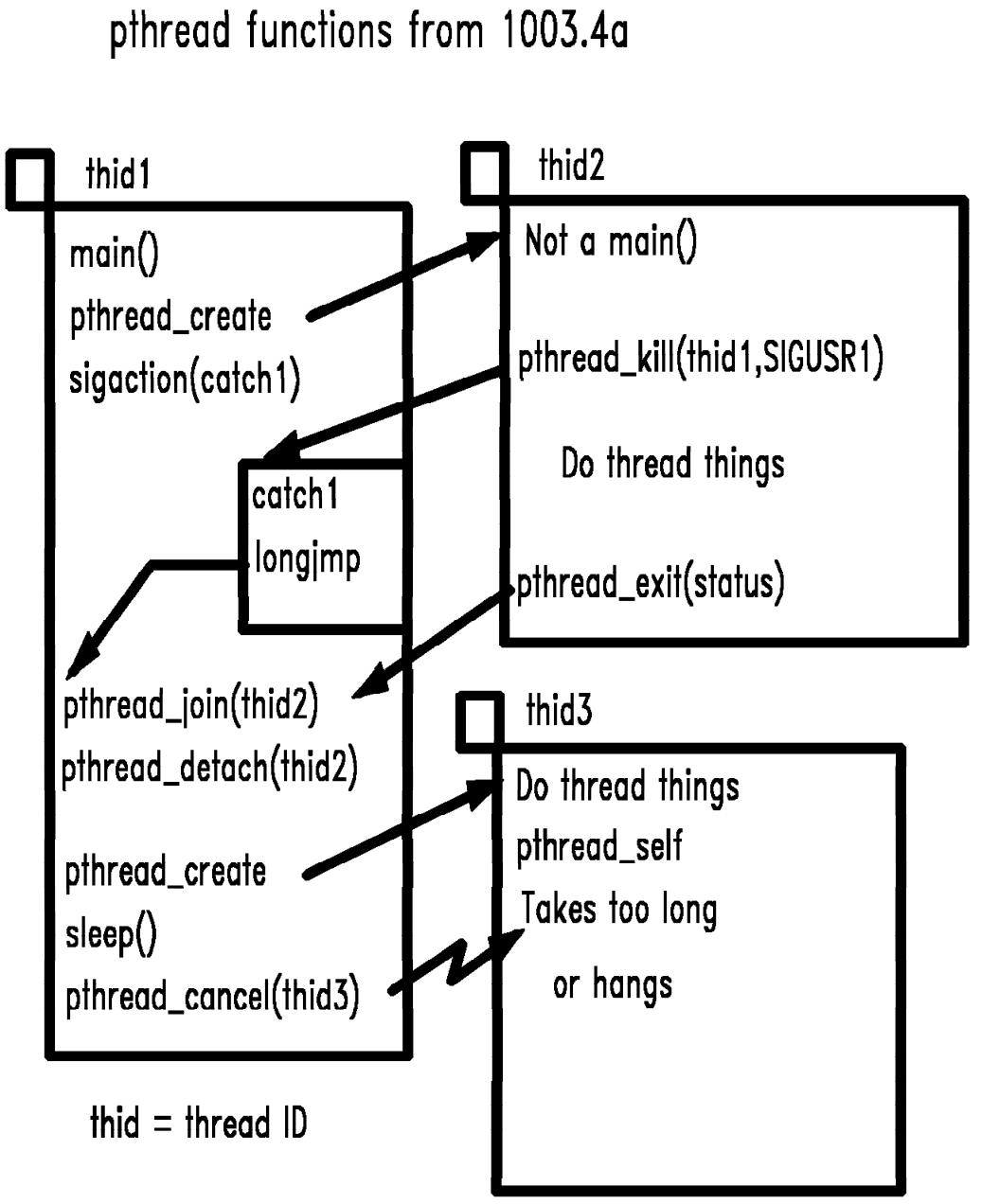


- LIBPATH environment variable defines HFS search order
- Supports multithreaded environment

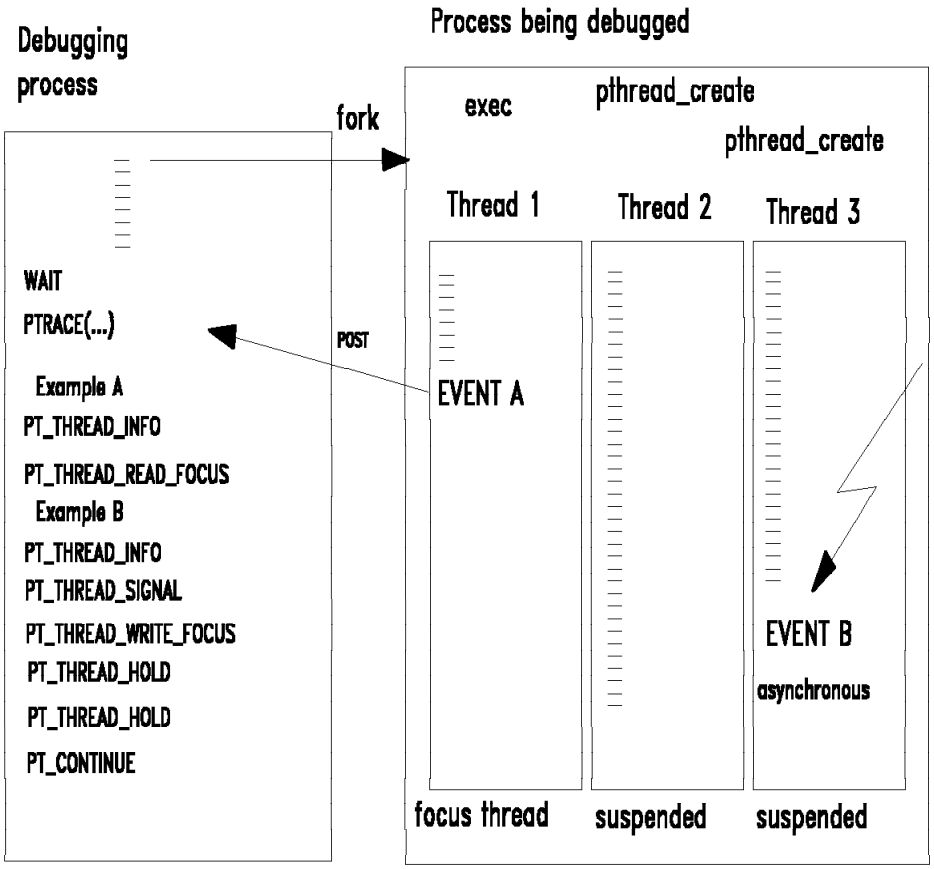
DLL

SIGNAL SUMMARY

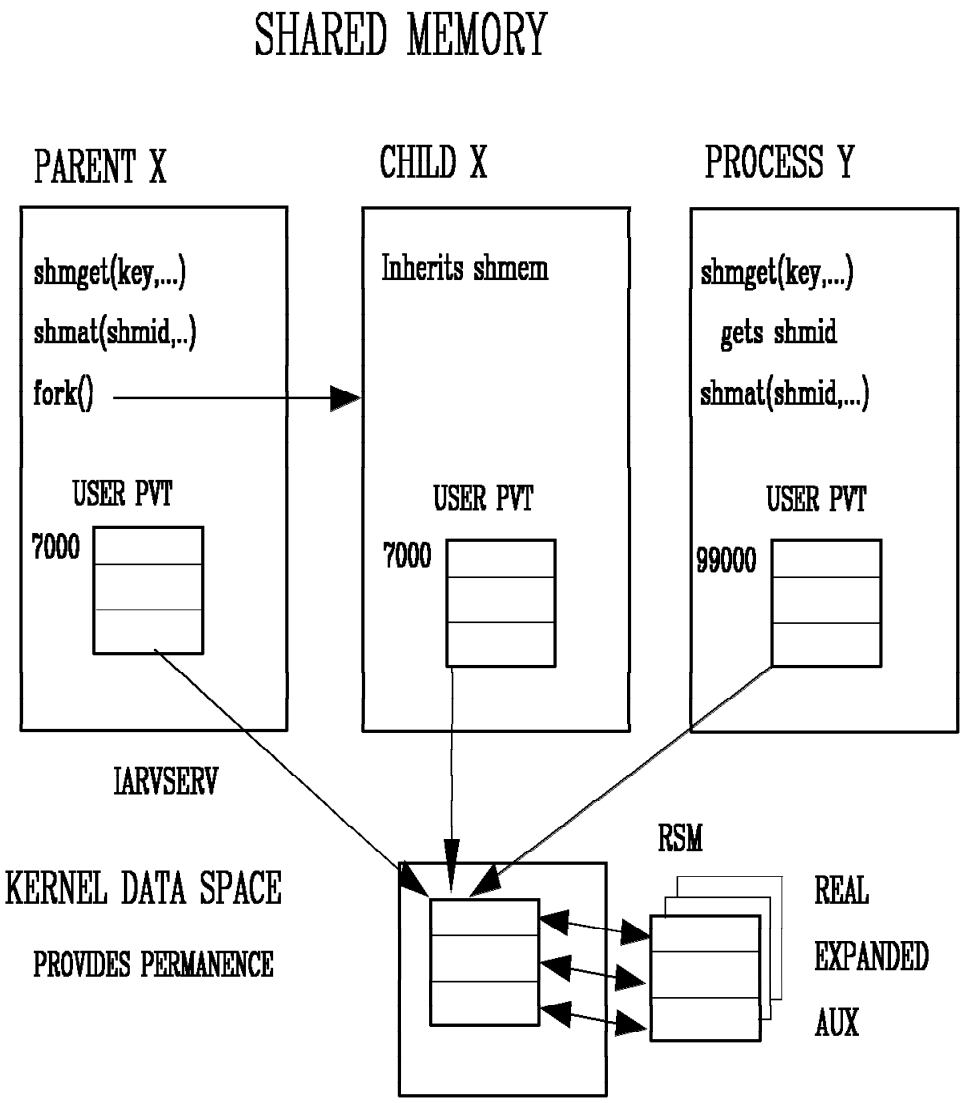



pthread functions from 1003.4a
thid1
main()
pthread_create
sigaction(catch1)
catch1
longjmp
pthread_join(thid2)
pthread_detach(thid2)
pthread_create
sleep()
pthread_cancel(thid3)
thid = thread ID
thid2
Not a main()
pthread_kill(thid1,SIGUSR1)
Do thread things
pthread_exit(status)
thid3
Do thread things
pthread_self
Takes too long
or hangs
pthread

New PTRACE and dbx Functions



DBXTHDS



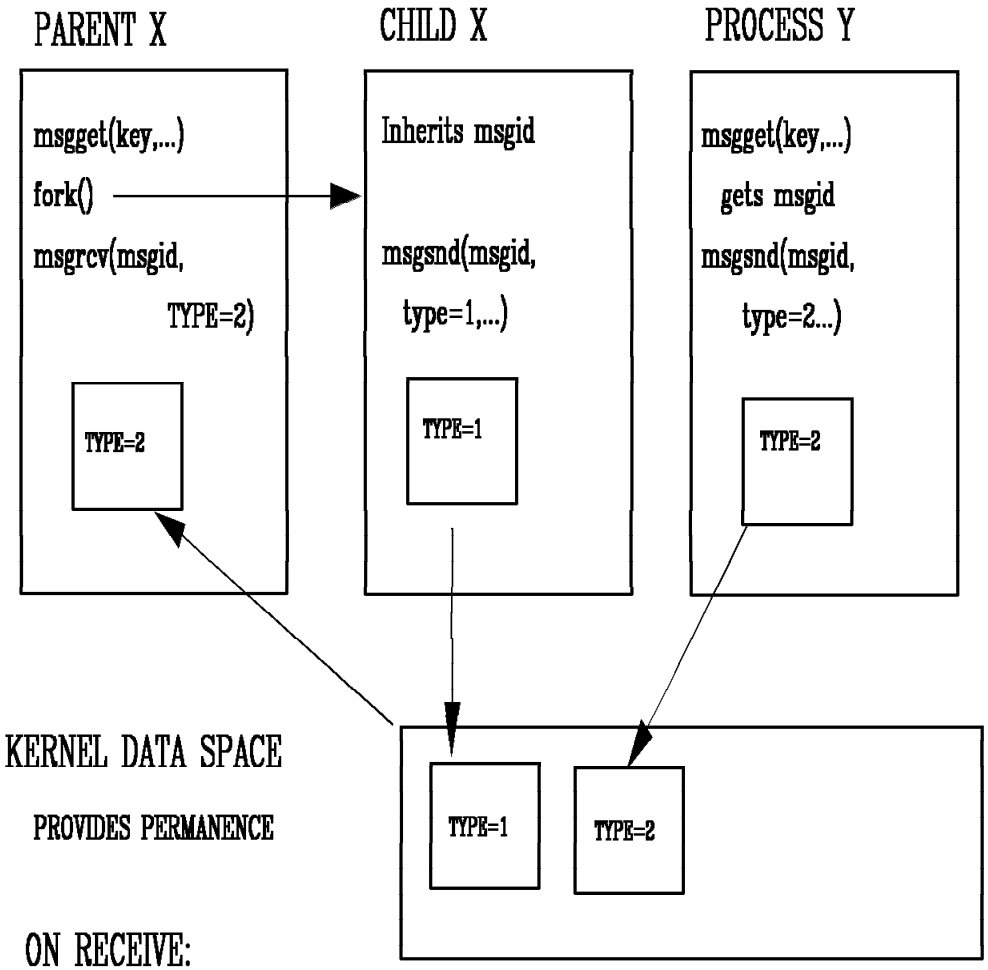
RSM overhead uses 32 bytes per connection/page.

Assume 10 users share 1 Meg:

10 users x 256 pages/Meg x 32 bytes/page = 82K of SQA

SHMEM

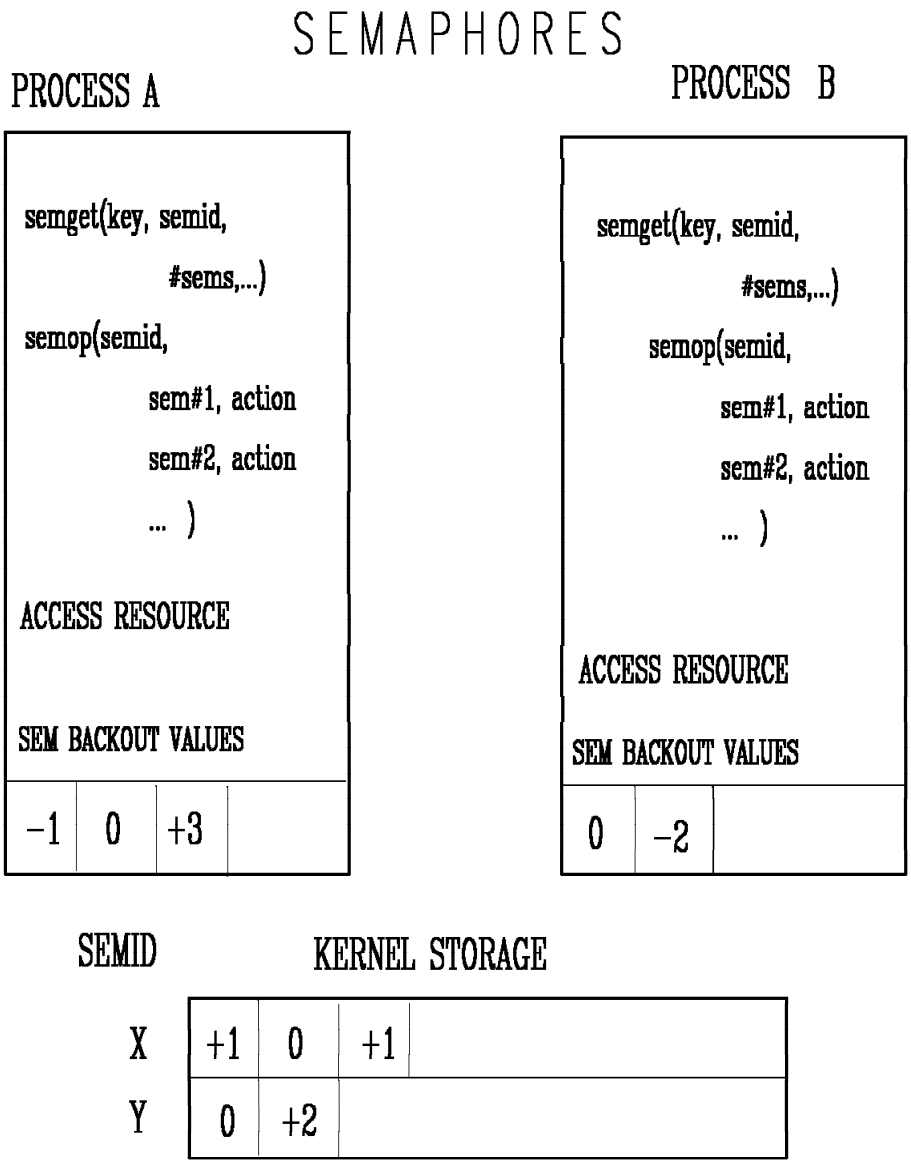
MESSAGE QUEUES



ON RECEIVE:

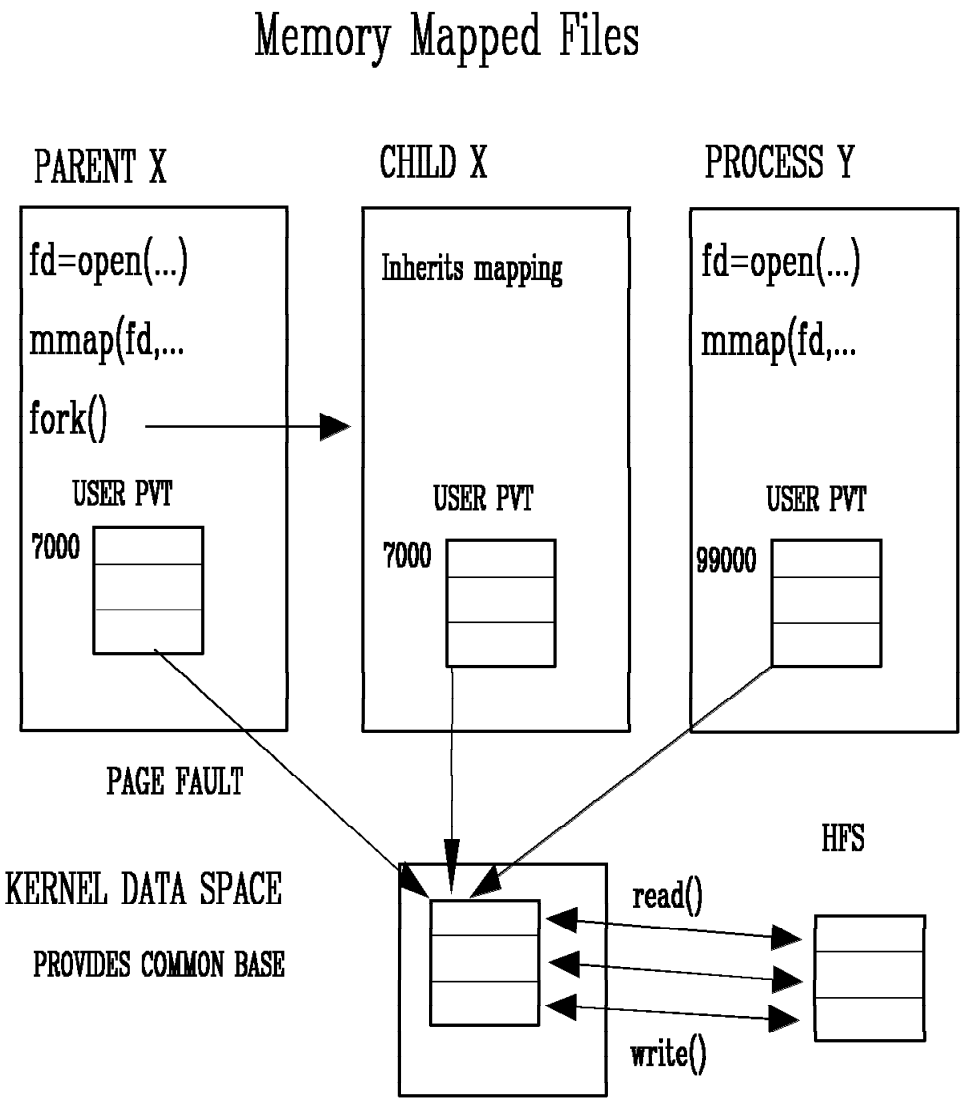
- Request a specific message type
- Request any of a certain type or lower
- Can request blocking or non-blocking

MSGQ



Each semop call can change or test 1 or more semaphores in a semaphore set atomically. Use to serialize shmem.

SEMA



RSM overhead uses 32 bytes of SQA per connection/page.

MMAP

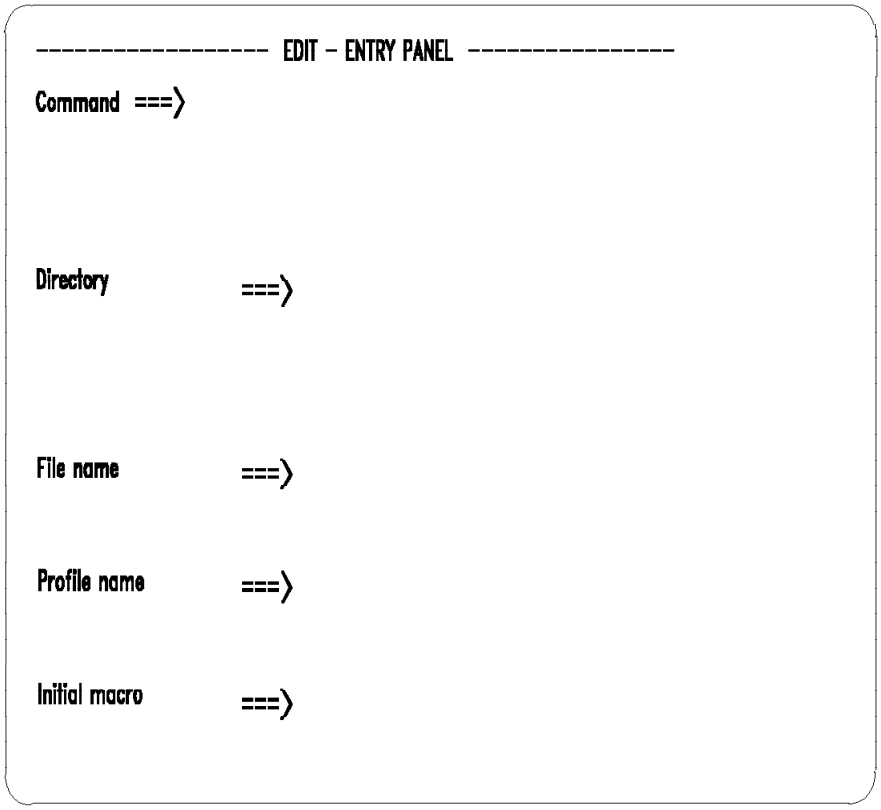
Execute program in file system

BPXBATCH invocation methods

- JCL EXEC statement
- TSO CALL command
- As a TSO command
- CLISTs or REXX EXECs

```
//jobname  JOB ...
//stepname EXEC PGM=BPXBATCH,PARM='SH cmd ...'
   OR
//stepname EXEC PGM=BPXBATCH,PARM='PGM name ...'
   OR
//stepname EXEC PGM=BPXBATCH
//STDIN    DD  PATH='/stdin_file_path'
//STDOUT   DD  PATH='/stdout_file_path',...
//STDERR   DD  PATH='/stderr_file_path',...
//STDENV   DD  PATH='/stdenv_file_path',...
  OR
//STDENV   DD  DSN=STDENV.DSNAME,...
  OR
//STDENV   DD  *
PATH=/bin
ENVVAR2=XXX
 ...
/*
```

EDITING AND BROWSING HFS FILES



10,'CMD(OBROWSE)'

20,'CMD(OEDIT)'

COPY
MOVE
REPLACE
CREATE
EDIT

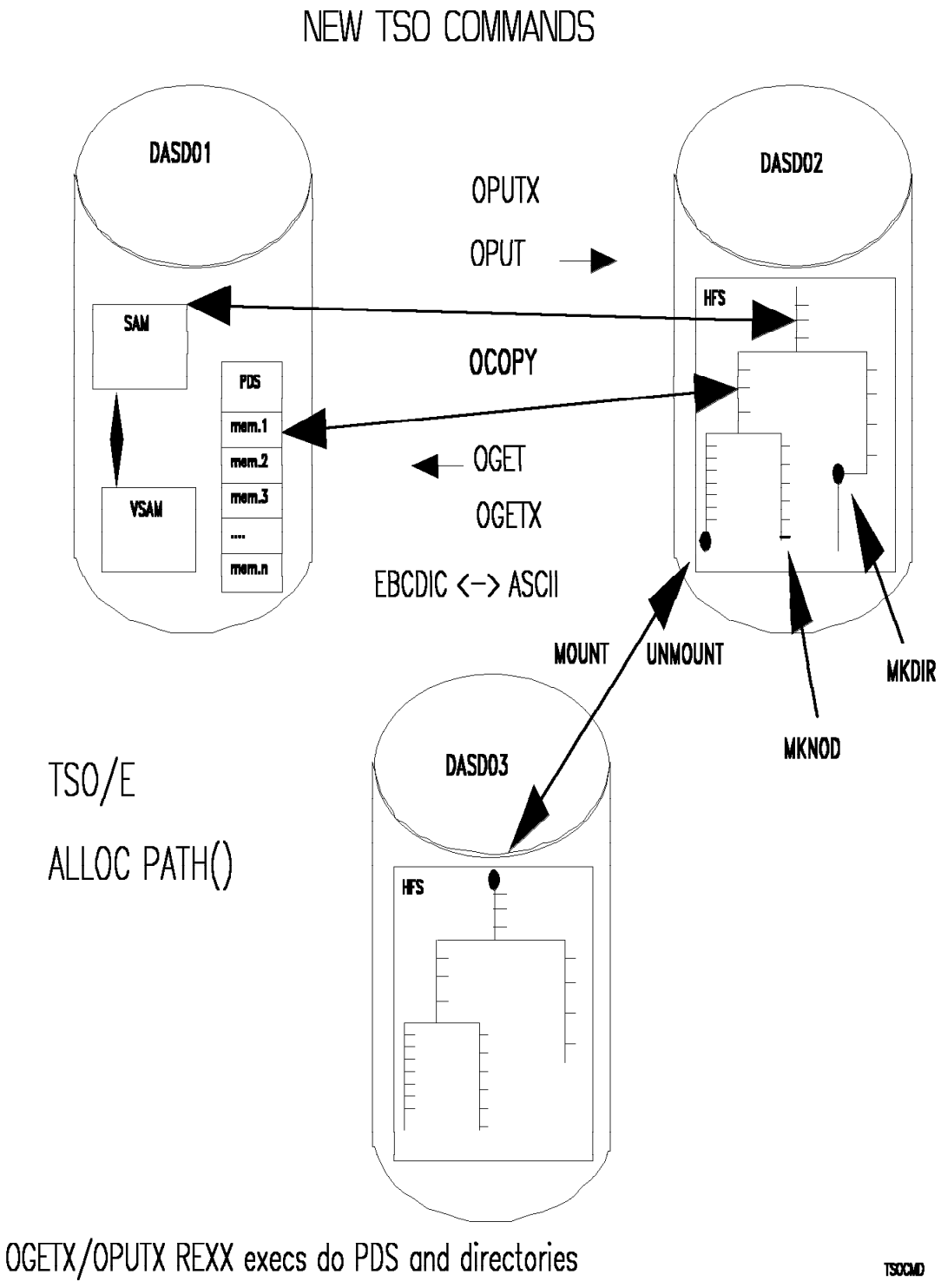
External Command

ISHELL contains many file management functions and systems management tools.

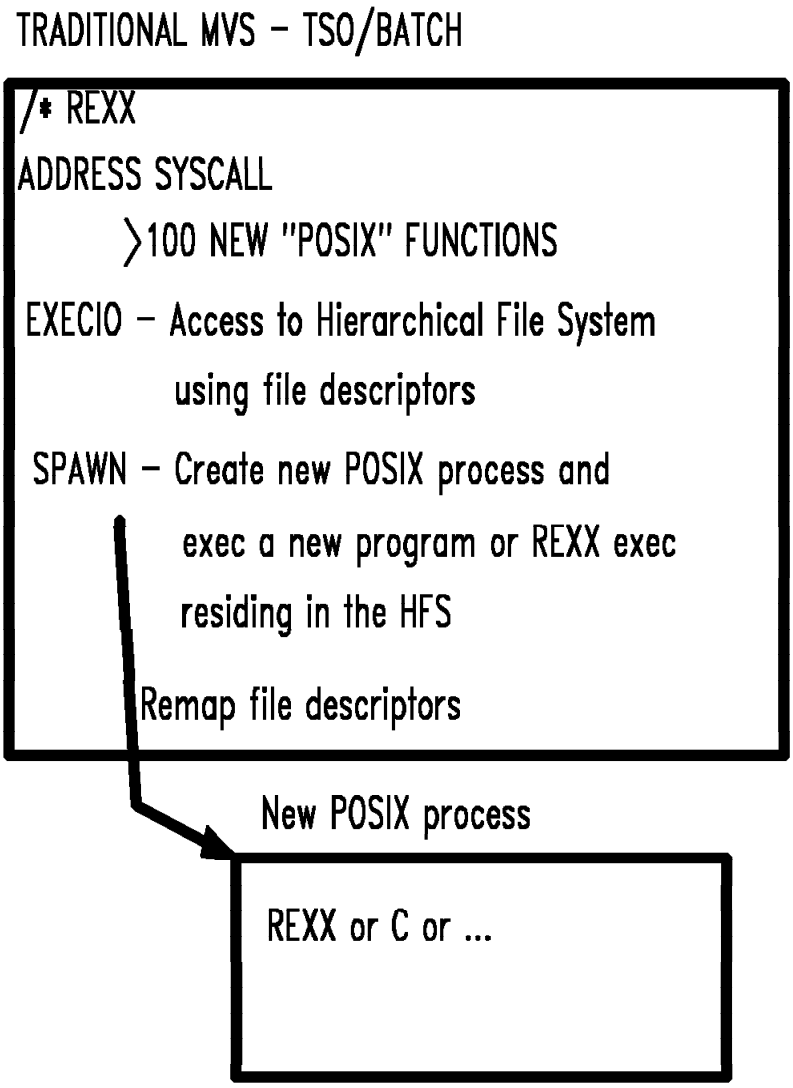
ISPF

```
File  Directory  Special_file  Tools  File_systems  Options  Setup  Help
 File
 1. New(N)...              Directory
 2. Attributes(A)...       1. List directory(L)...
 3. Delete(D)...           2. New(N)...                1. *User...
 4. Rename(R)...           3. Attributes(A)...         2. *User list...
 5. Edit(E)...             4. Delete(D)...             3. *All users...
 6. Browse text(B)...      5. Rename(R)...             4. *All groups...
 7. Browse records(V)      6. Copy to PDS(C)...        5. *Permit field access
 8. Copy to(C)...          7. Copy from PDS(I)...      6. *Character Special..
 9. Replace from(I)..      8. Print(P)                 7. *Root...
10. Print(P)               9. Compare(M)...            8. Enable superuser mod
11. Compare(M)...         10. Find strings(F)...        (*) require superuser
12. Find strings(F)..     11. Set working directory
13. Run(X)...             12. File system(U)...
14. Link...
15. File system(U)...
```

TSO commands



REXX ENHANCEMENTS



REXX1

REXX ENHANCEMENTS

TRADITIONAL MVS - TSO/BATCH or SHELL

```
/* REXX
ADDRESS SH
        ACCESS TO SHELL & UTILITIES
REXX support remaps file descriptors
 SAY 'HOWDI' - Mapped to STDOUT fd=1
 PARSE EXTERNAL - Mapped to STDIN fd=0
 EXECIO - access to HFS
```

Allows mixing of shell commands and REXX
with output piped between them.
Allows multitasking REXX connected with pipes.

REXX2

File System

Hierarchical directories
(Current working directory)

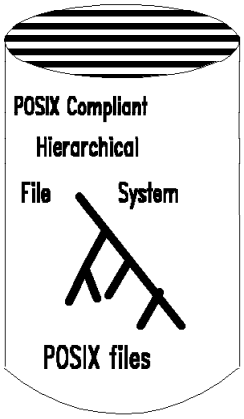
All data treated as byte streams
(application applies structure)

Concurrent write to same file
from multiple address spaces

Permission control

Utilities

Sparse Files



POSIX syscall semantics

Byte range locking
(voluntary)

Long file names

Symbolic links

Pipes

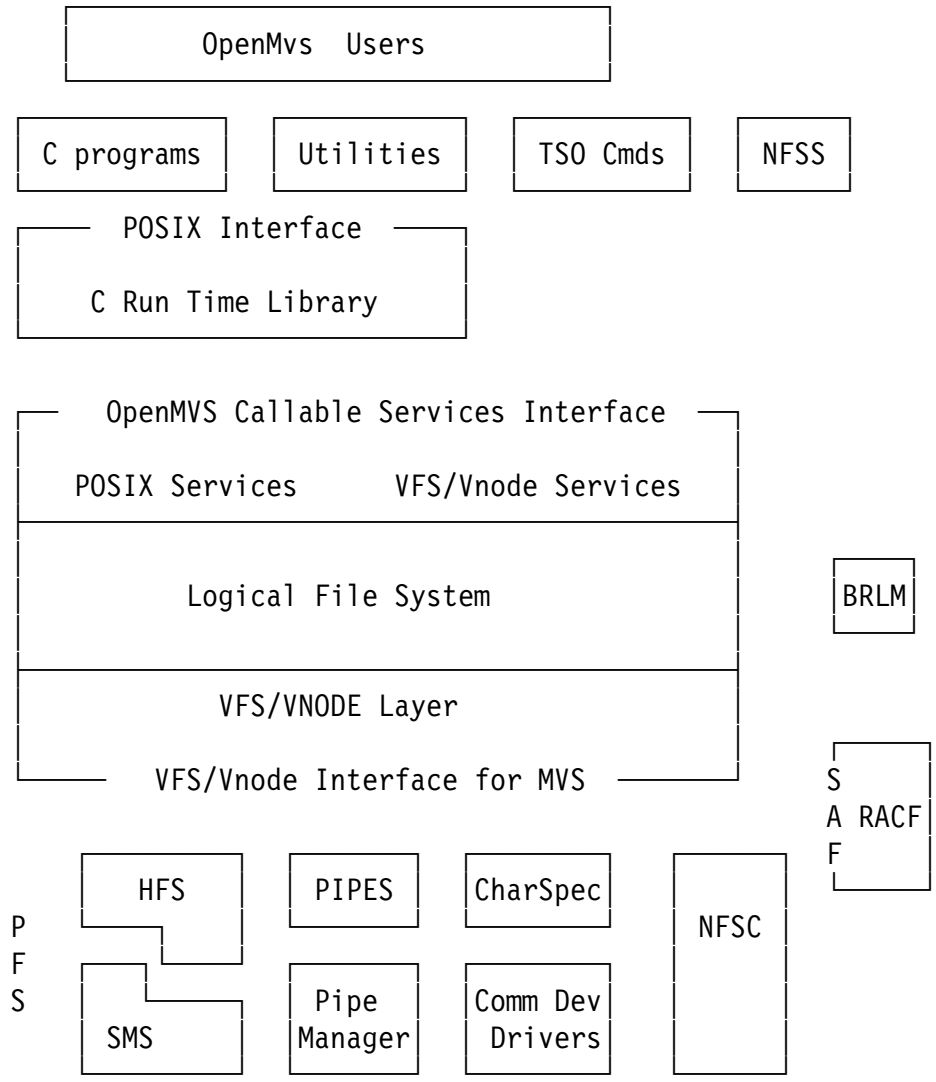
HSM support

ADSM

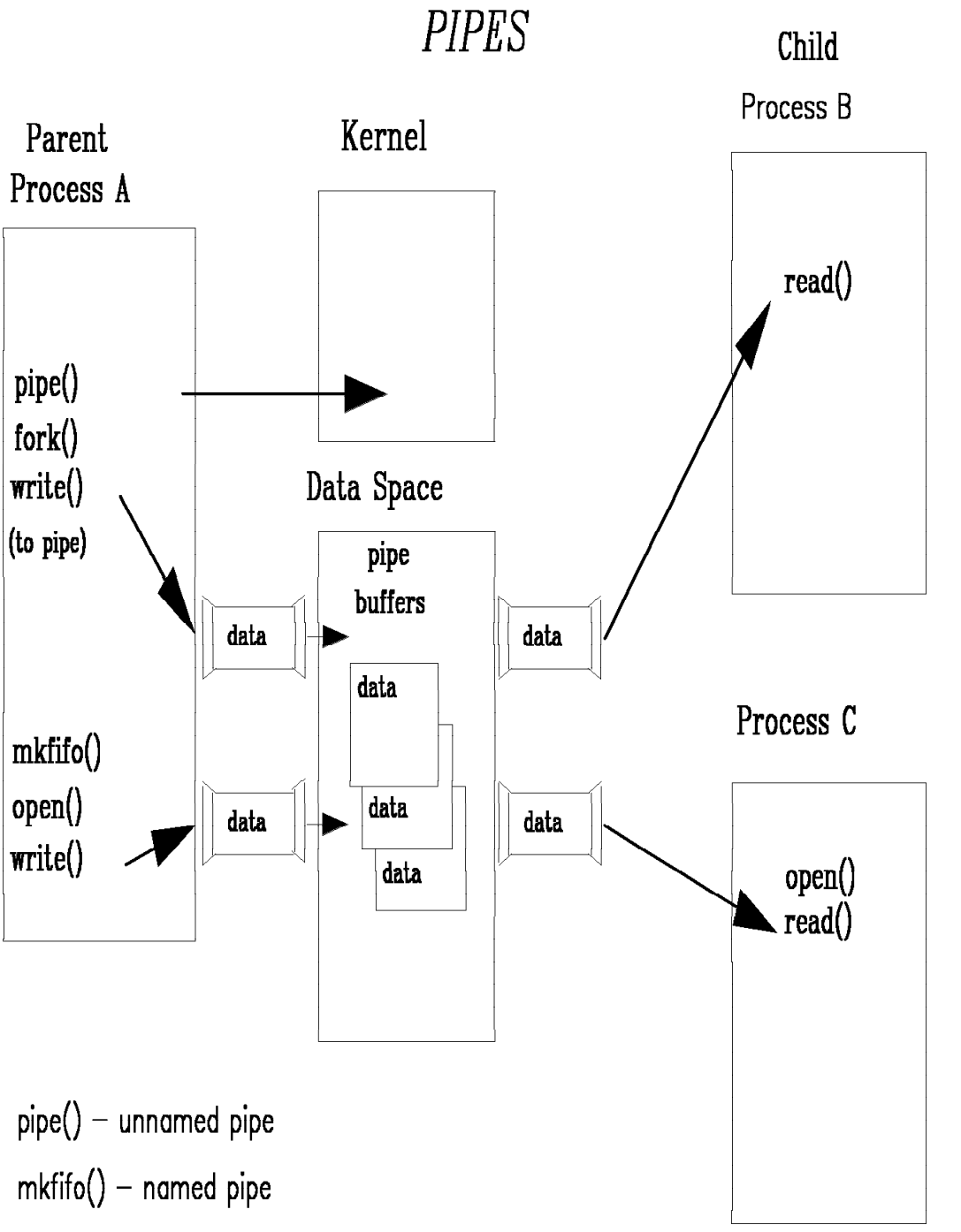
HFS SOCKETS PTY RTY NFSS DFSS

FILESYS1

System Layers



BRLM: Byte Range Lock Manager
HFS: Hierarchical File System
LFS: Logical File System
PFS: Physical File System
VFS: Virtual File System


PIPES
Child
Process B
Parent
Process A
Kernel
read()
pipe()
fork()
write()
(to pipe)
Data Space
pipe
buffers
data
data
data
data
data
data
mkfifo()
open()
write()
Process C
open()
read()
pipe() – unnamed pipe
mkfifo() – named pipe
PIPES1

File System ...

Hierarchical File System (HFS)

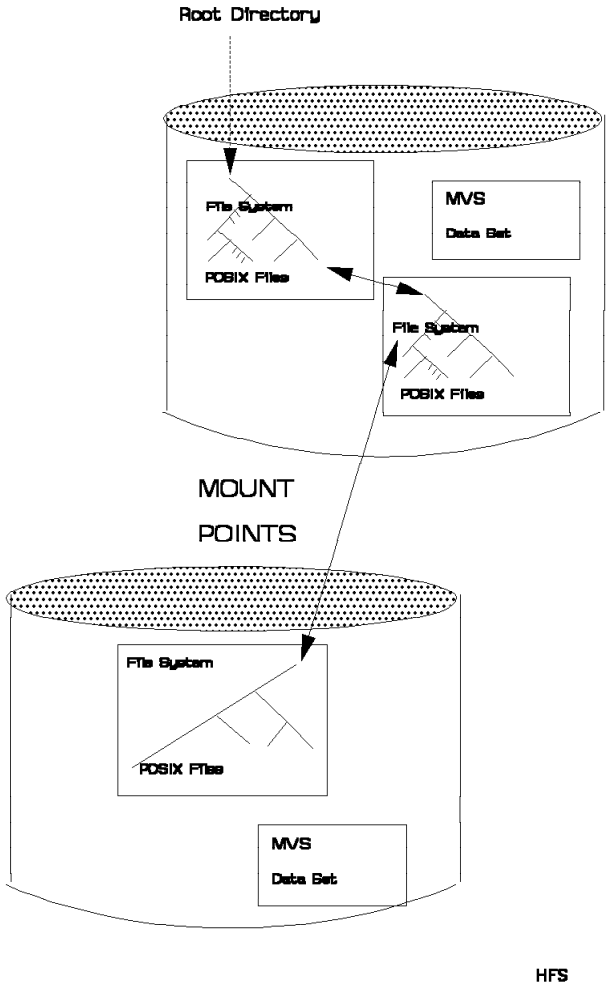
Single: Hierarchical File System

- POSIX 1003.1 Compliant
- Single root directory

Multiple: Mountable File Systems

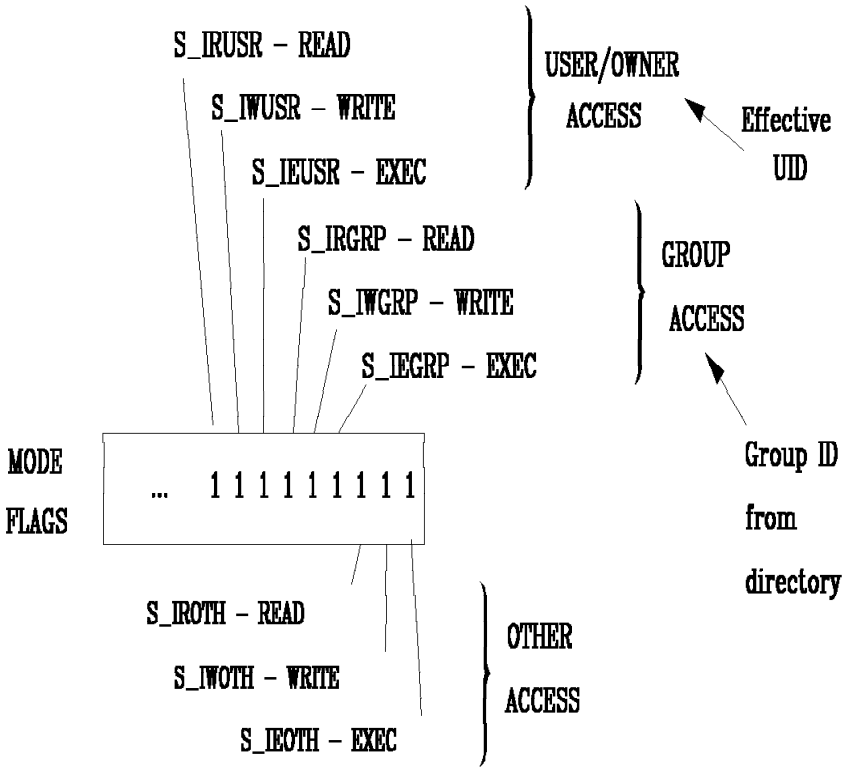
- Mounted at arbitrary directories
- A new MVS dataset type
- Managed by DFSMS

```
//DD1 DD DSNTYPE=HFS,
// DSN=SYS1.ROOT,
// STORCLAS=ALL,
// DISP=(NEW,KEEP),
// SPACE=(CYL,(10,5,1))
```



HFS

FILE ACCESS CONTROL



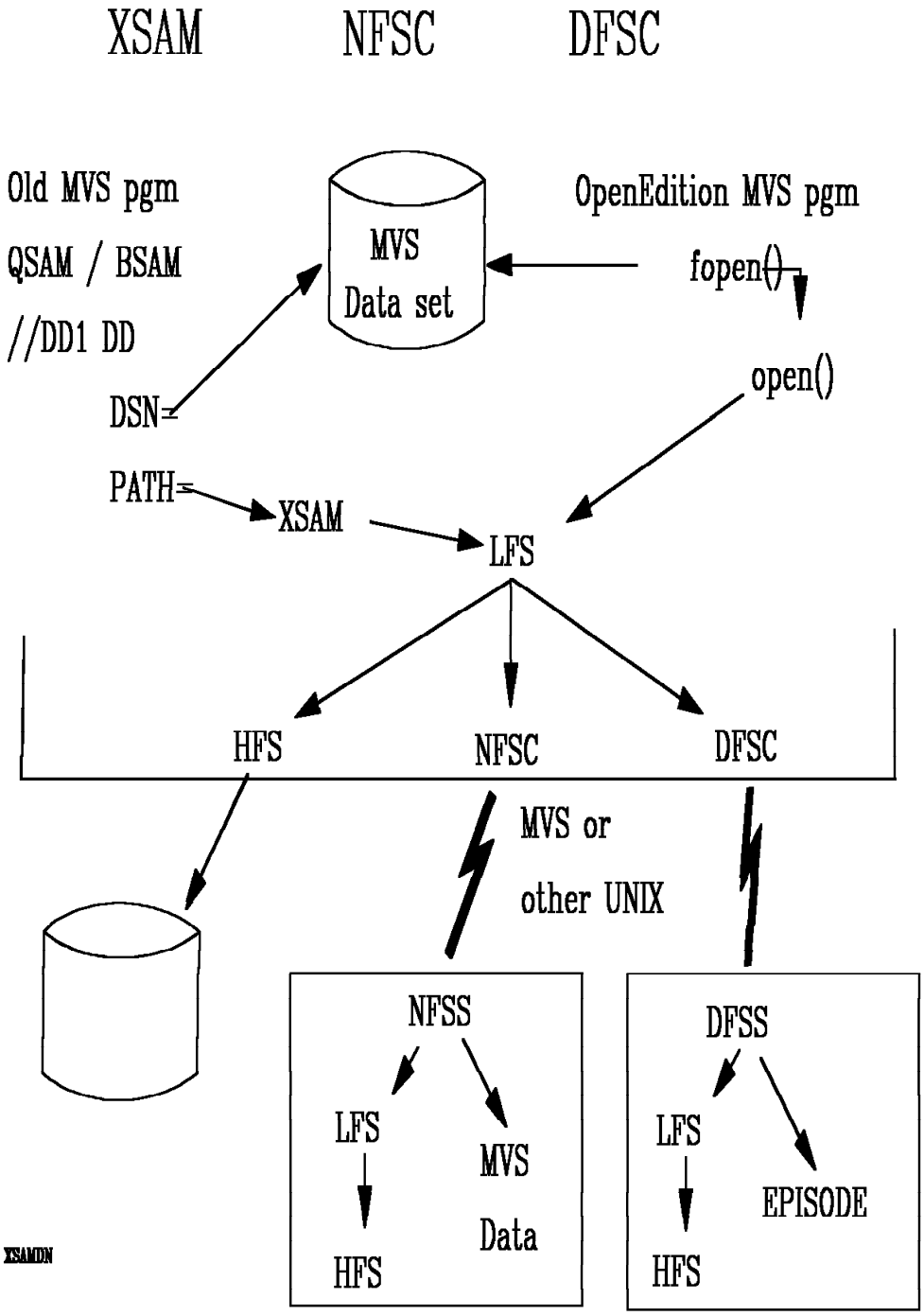
Permissions specified on JCL (PATHMODE=), BPX1OPN, or BPX1CHM (chmod).

BPXYMODE contains mode map and constants.

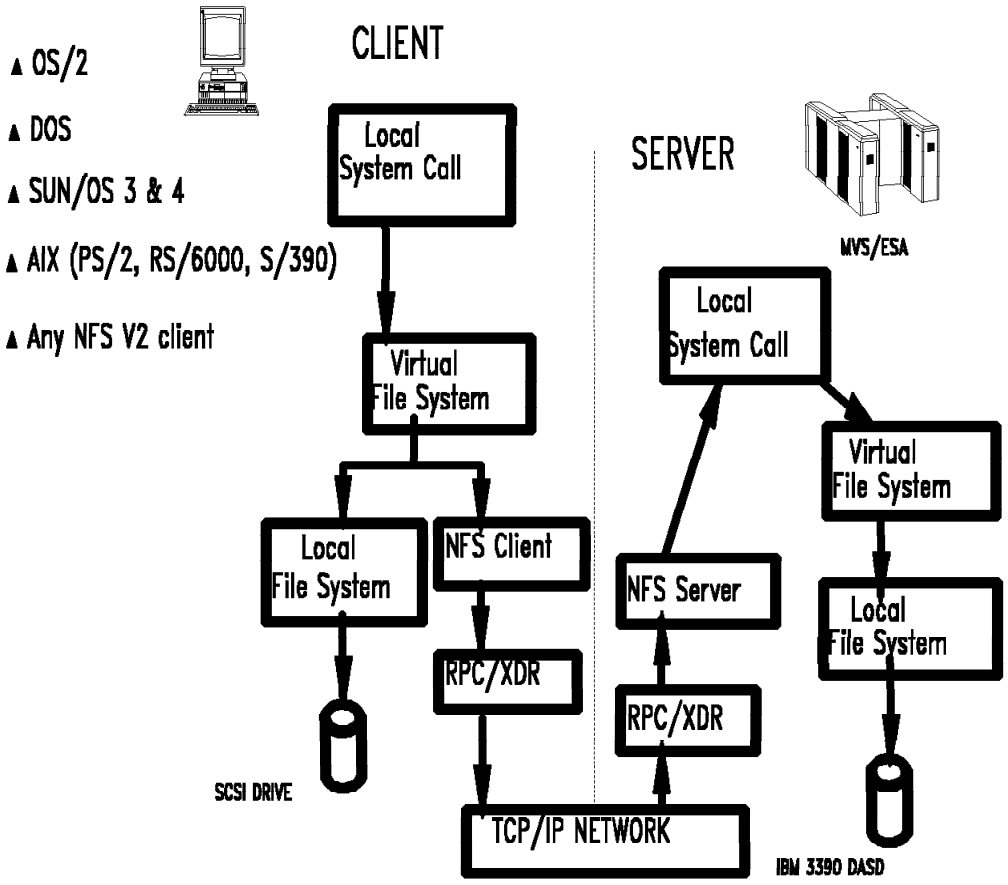
SETUID, SETGID, STICKY

PERMISS

XSAM/DFSC/NFSC



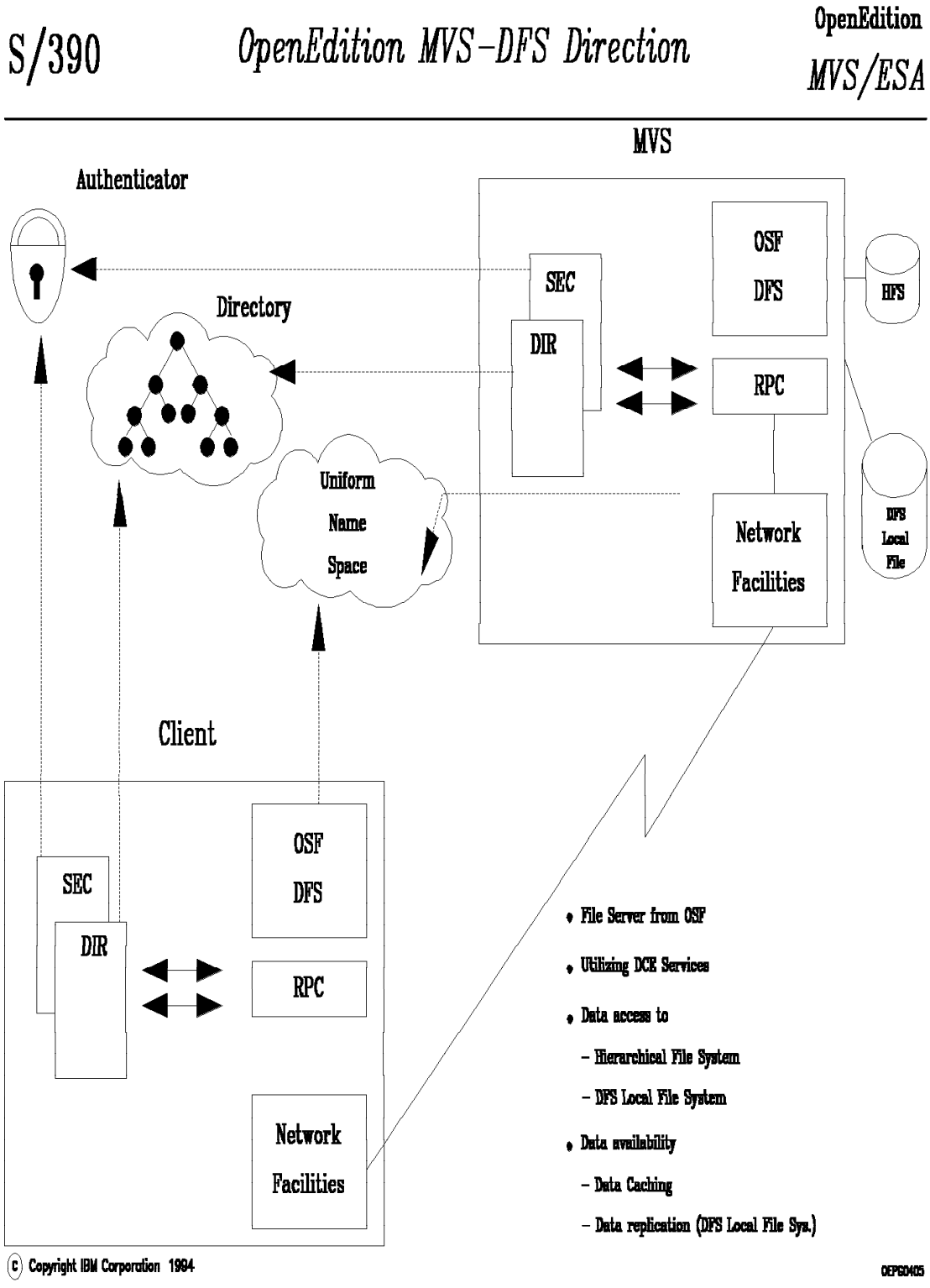
OpenEdition Support for NFS



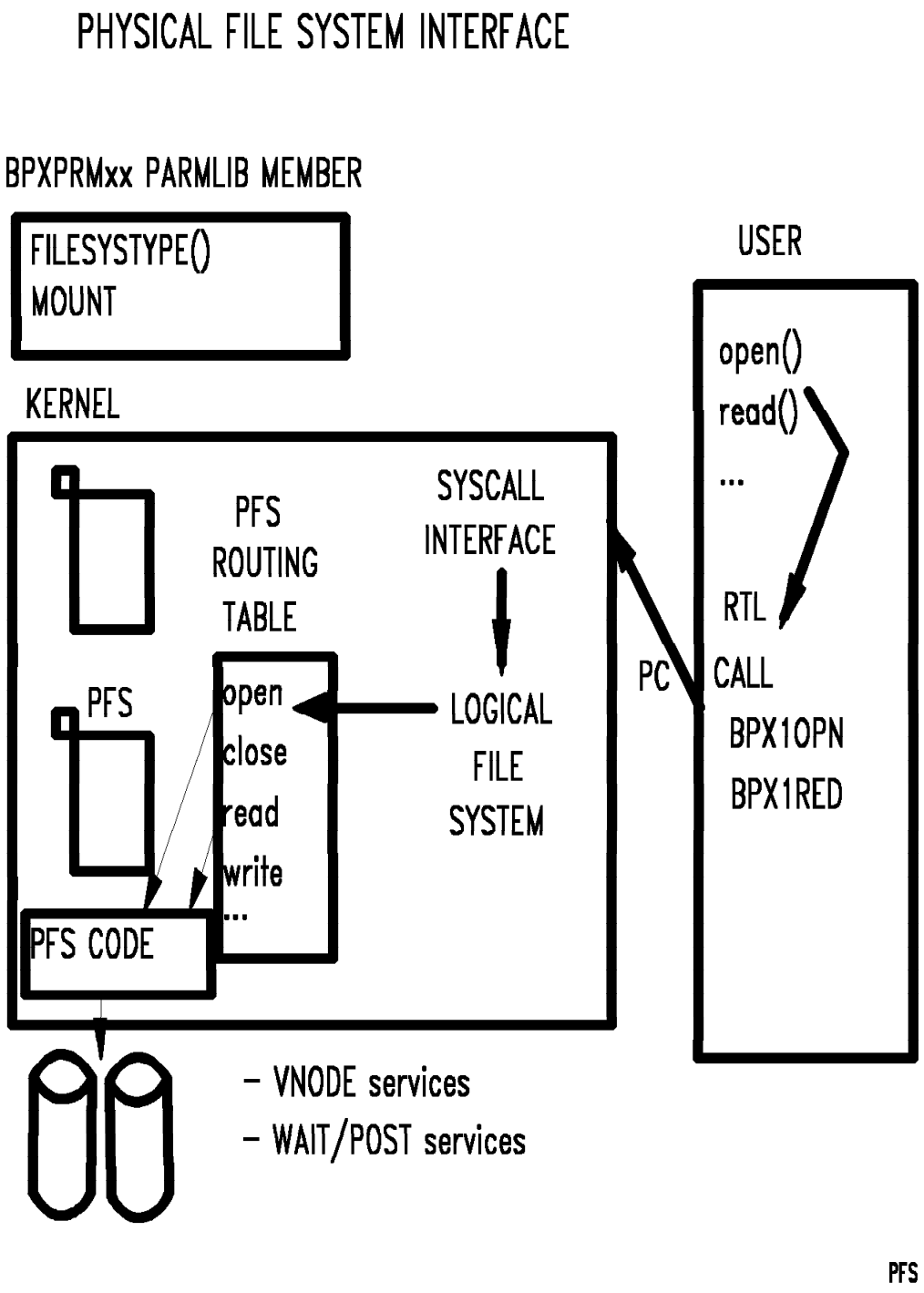
- Provides distributed data access
- Widely used across many system platforms
- Supports access to legacy data

NFSS

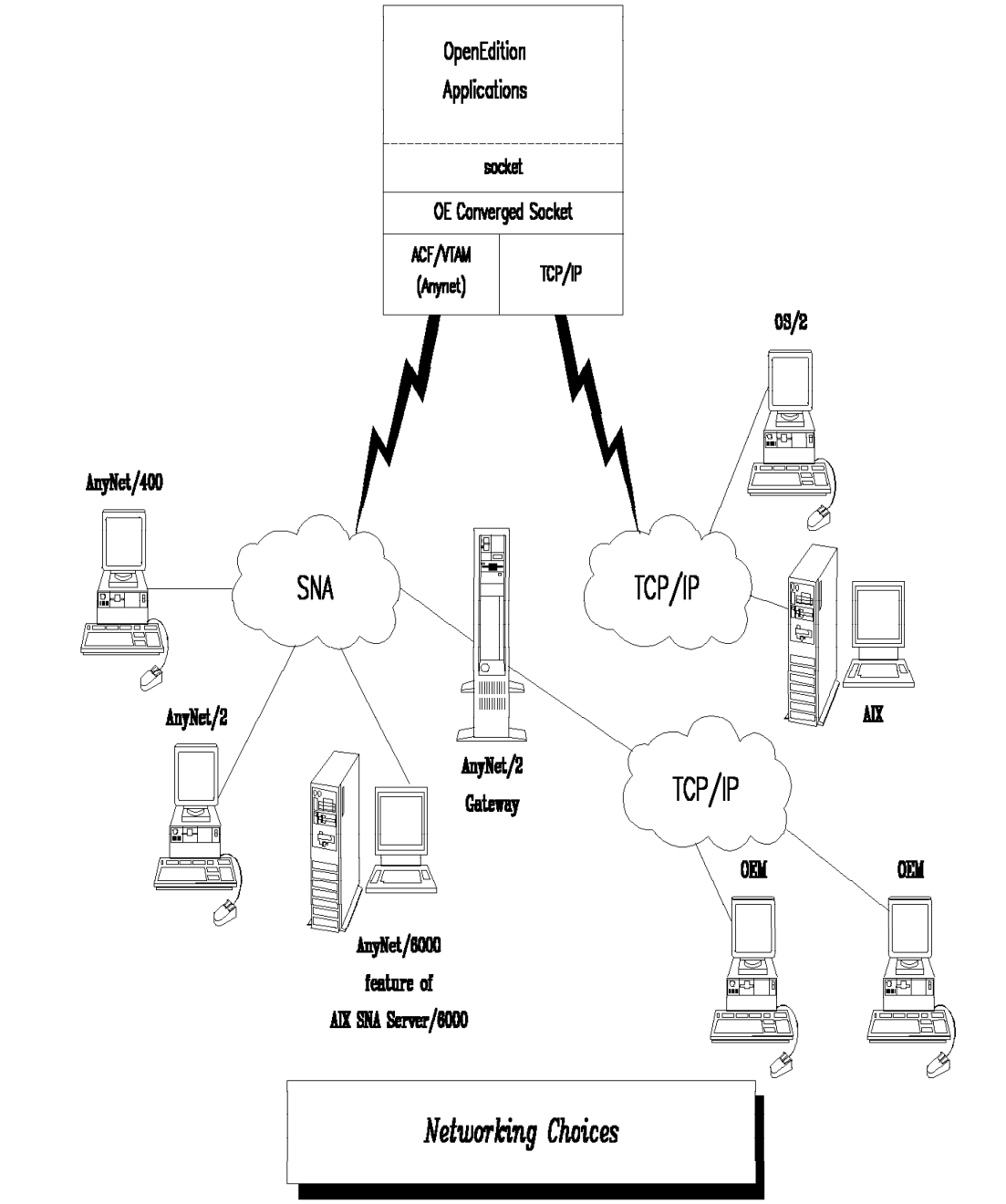
File Server Enablement ...



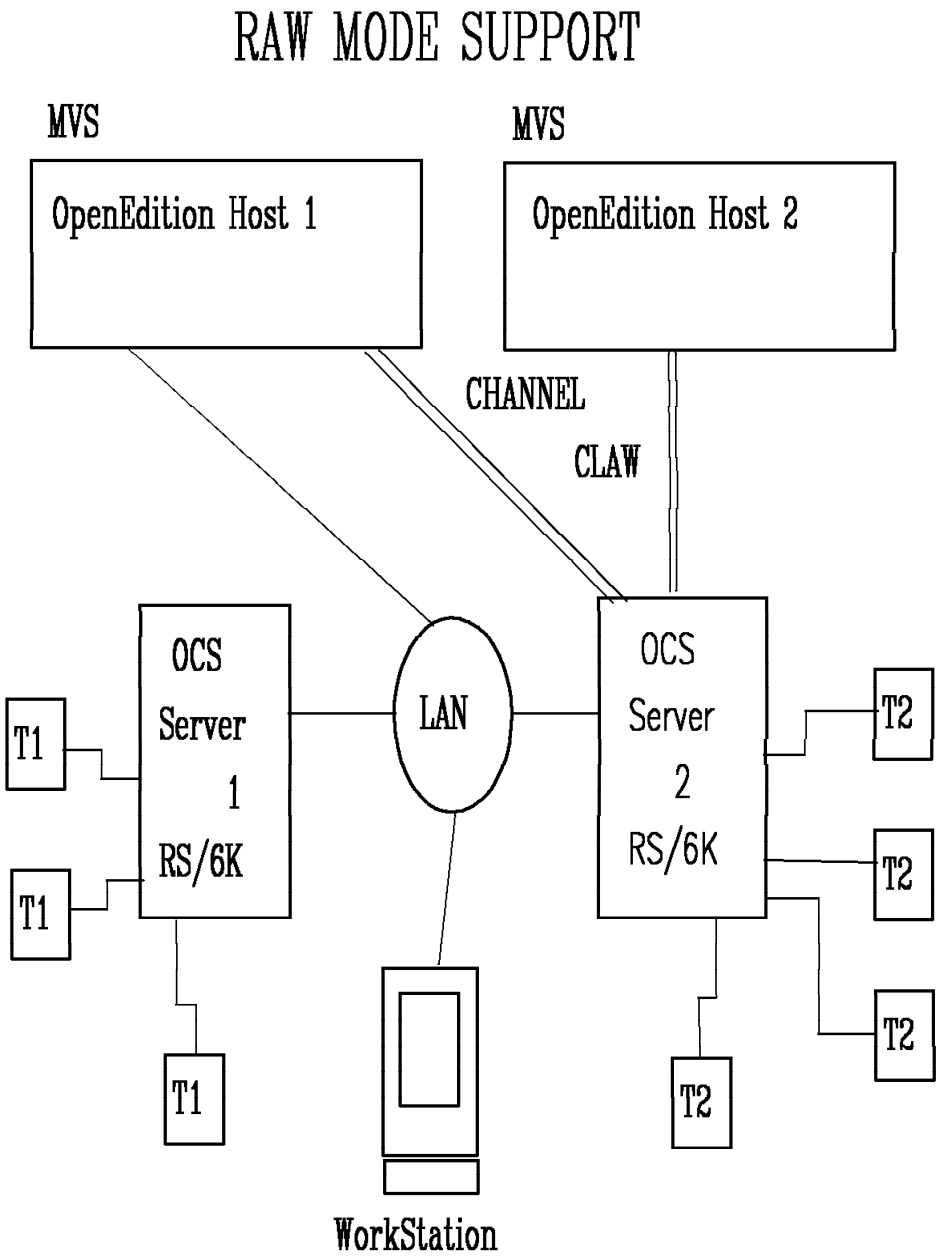
File Server Enablement ...



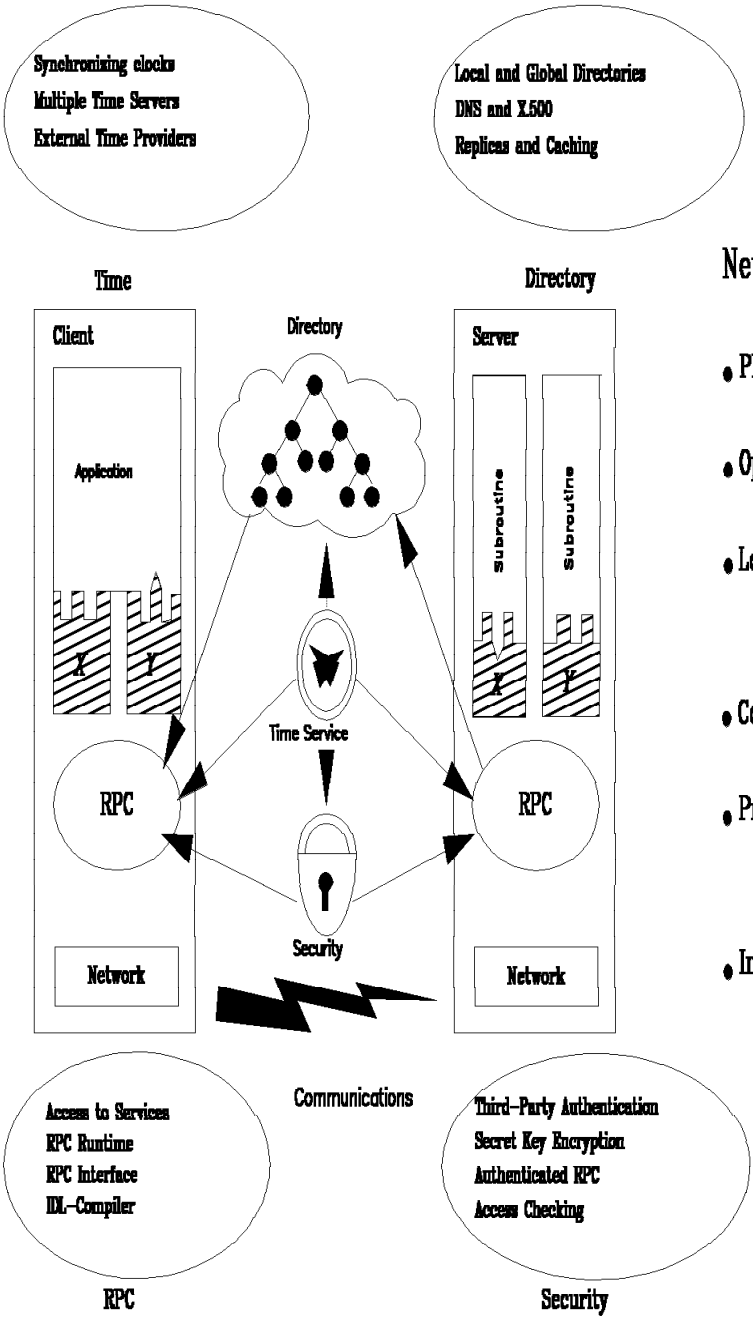
S/390 — *SNA Networking for 'socket' Applications* — OpenEdition MVS/ESA



OEPG0358


RAW MODE SUPPORT
MVS
MVS
OpenEdition Host 1
OpenEdition Host 2
CHANNEL
CLAW
OCS
Server
1
RS/6K
LAN
OCS
Server
2
RS/6K
T1
T1
T1
T2
T2
T2
T2
WorkStation
Bytes clustered in OCS Server and delivered on the host
OCSRAW

S/390 *Distributed Computing Environment* OpenEdition *MVS/ESA*

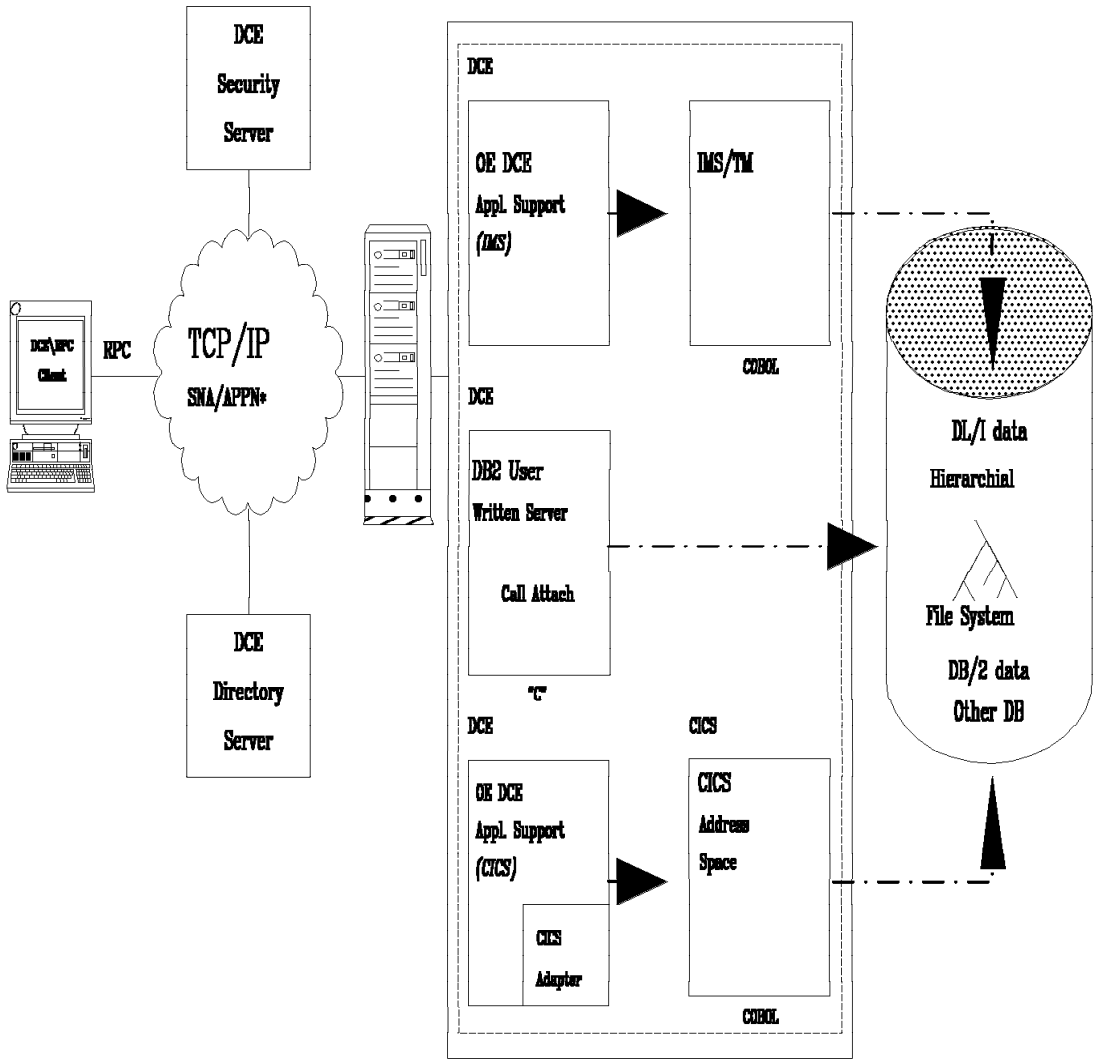


Network Services

- Platform independent
- Open technology based
- Locate resources throughout the network
- Control user access
- Provides platform for shared business logic
- Increases programmer productivity

OEPG0370

S/390 — *MVS/DCE Server Access* — OpenEdition *MVS/ESA*

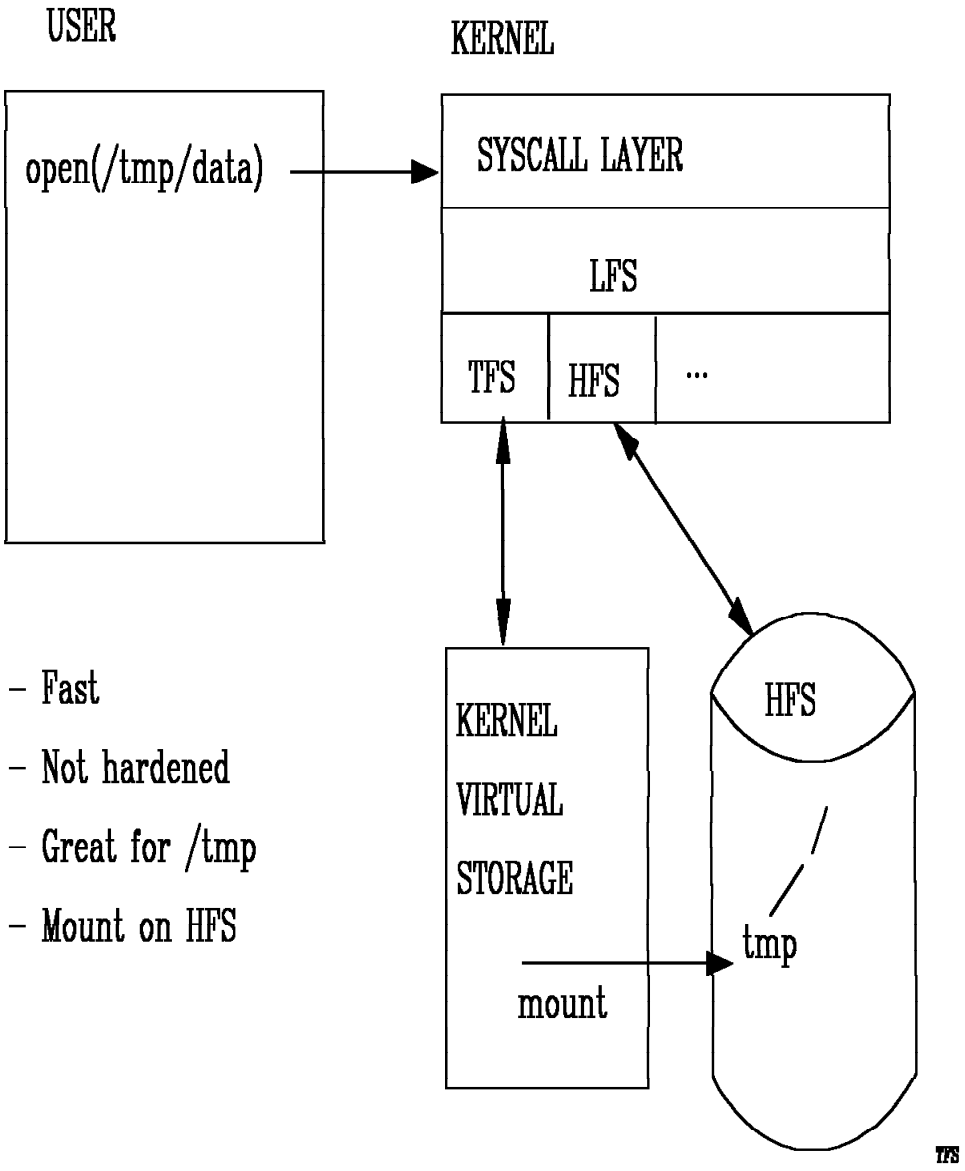


MVS DCE Servers

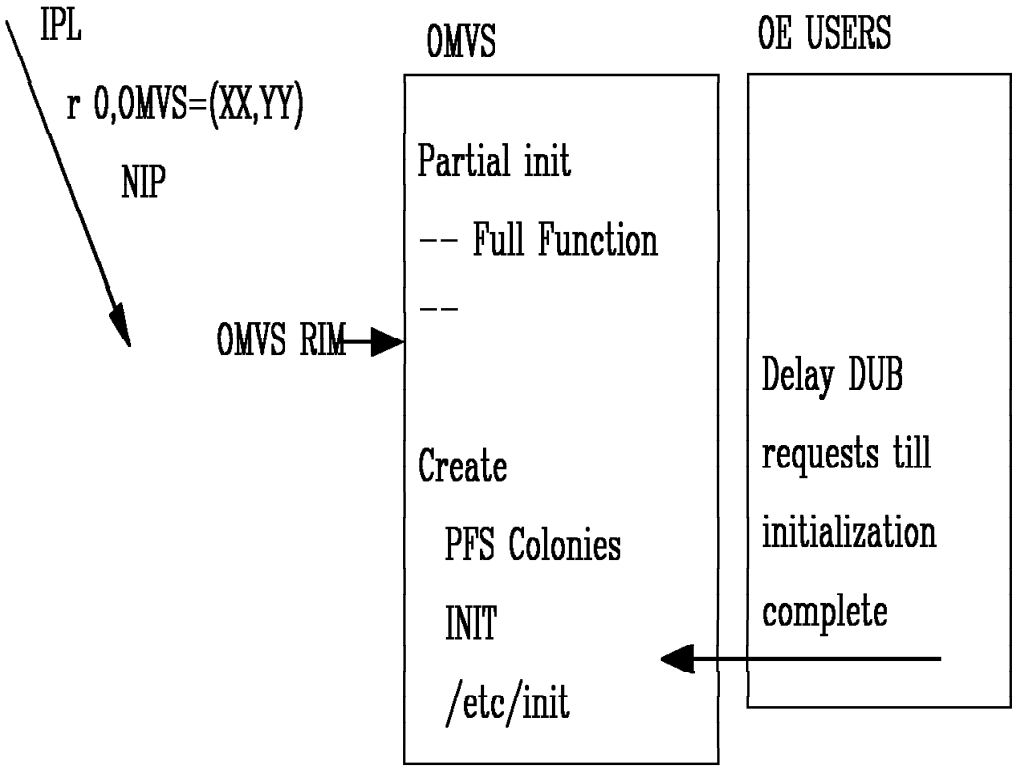
* SNA/APPN Support Planned

OEPG0360

Temporary File System (TFS)



PERMANENT KERNEL



SET OMVS=(xx,yy,..) Pick up new BPXPRMxx,yy

SETOMVS options to change selected kernel parms

D OMVS support to display current option settings

P OMVS no longer supported

PERMKERN

Permanent Kernel ...

This was a tough decision but was made about 2 years ago. In a nutshell:

1. Vendors porting code to OS/390 do not deal with ″kernel termination″. It is not a concept on UNIX machines, therefore their ported code does not deal with it.
2. Numerous projects within IBM refused to base their product on OE unless it was permanently part of the operating system. This included TCP/IP, DB2 and CICS.
3. OE provides basic operating system functions. I compare it to VSM. If VSM breaks and no one can do a GETMAIN, then the system is dead and unusable, so it needs to be reIPLed. From a UNIX perspective, if they call open(), it either works or it doesn′t. The user can′t deal with ″kernel is down″. What should they do about it? They can′t wait, since the kernel provides the wait service.
4. Some folks say that if OE goes down, they can′t afford a reIPL because it will take down their bread and butter applications (DB2/CICS). DB2 is already using OE and CICS will soon. Neither of them are willing to write code to deal with kernel termination.
5. Our intent is for OE to be as robust as the rest of the operating system. On this front we have probably not succeeded as well as we would like. OE availability is the sum of the availability of the kernel, file system (SMS), TCP/IP and the servers that run on OE. As for the kernel, we rarely see a reIPL reason that the kernel is responsible for. Applications on the other hand can hang up a significant number of users. This could happen with existing MVS applications, but it doesn′t because MVS applications tend to be single address space in orientation. Many UNIX servers are heavily multiprocess in orientation. Their concept of robustness leaves something to be desired by OS/390 standards. We treat any problem that forces a reIPL as something that needs to be fixed.
6. By making the kernel a permanent part of the OS/390 operating system, it has allowed us to make dramatic improvements in the performance of the kernel calls. We have been able to

reduce the syscall layer from 500 instructions to 100 instructions.

7. If customers really want to shut down OE and restart it, we could FORCE all OE users through memterm and then restart OE. Of all the customers I've talked to, none seemed willing to accept the FORCE as the solution.
8. If we had initially created OMVS as a permanent part of the system, then people would not be complaining now. The fact that they could mess up the system, P OMVS, S OMVS was very convenient while OE was simply a toy. Since OE is now being used for real production work, this method of operation is no longer viable.

But the kernel is working on a design to allow the following:

- Restarting the JES subsystem.
- Adding new filesystem types.
- Correcting spelling errors in file system startup parameters.
- Reestablishing a 'known state' for Unix System Services, ability to 'bounce'; useful for recovering from file system hangs.
- Controlled shutdown of Unix System Services, prior to System Re-Ipl.

Console communications (R2)

CONSOLE COMMUNICATIONS - R2

__console() - Calls BPX1CSS Callable service

- Listen for operator modify and stop commands
- Issue WTO to operator

F jobname.id,APPL=parms_to_pass_to_application

P jobname.id,A=asid

pthread_tag_np() - Calls BPX1PTT Callable service

- Tag a thread with a character string

display omvs,pid=123456

Displays all threads with state and TAG data.

Modify command can be used to cancel threads.

- Termination control without CANCEL:

F BPXOINIT,TERM=pid.tid (or just pid)

F BPXOINIT,FORCE=pid.tid

CONSOLE

SMF RECORDING - R2

__smf_record() calls BPX1SMF Callable Service

- Requires permission to BPX.SMF FACILITY class profile

- Allows C programs to write SMF records

- Exploitation planned by WEB

- Records up to 32K with SMF TYPE and SUBTYPE

SMFREC

Other R3 changes

- Tag service - string associated with a thread
- D OMVS service to display thread level info
- _BPX_ACCOUNT environment variable

Default UID/GID - OW27564

```
ADDGROUP OEDFLTG OMVS(GID(777777))

ADDUSER OEDFLTU DFLTGRP(OEDFLTG) NAME(¢OE DEFAULT USER¢)
        OMVS(UID(999999) HOME(¢/¢) PROGRAM(¢/bin/sh¢))

RDEFINE FACILITY BPX.DEFAULT.USER APPLDATA(¢OEDFLTU/OEDFLTG¢)
SETROPTS RACLIST(FACILITY) REFRESH
```

During DUB processing, the security product retrieves the default OMVS segments for user or group if OMVS segment for the user does not exist.

WLM interfaces in C - OW27544 (R4)

C __wlm -> BPX1WLM -> IWMxxxxx services

Permission to BPX.WLMSERVER FACILITY class required.

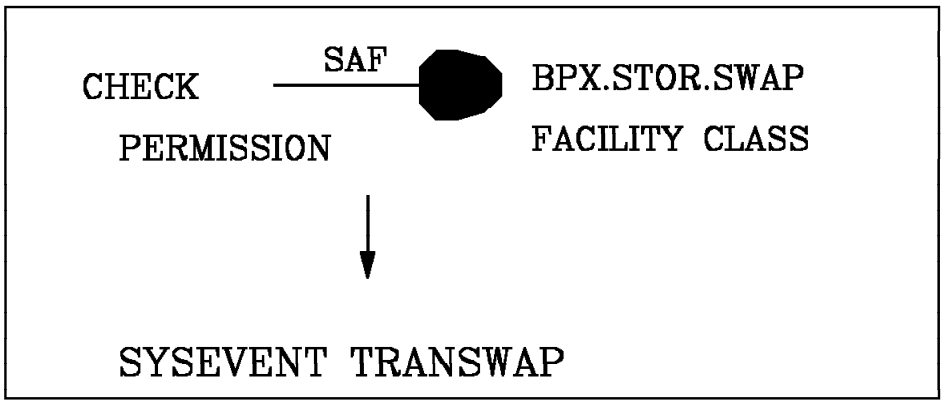
- Query WLM Scheduling Environment
- Check WLM Scheduling Environment
- Disconnect from WLM
- Delete a WLM Work Unit
- Join a WLM Work Unit
- Leave a WLM Work Unit
- Connect to WLM as a work manager
- Connect to WLM as a server manager
- Create an independent WLM work unit
- Create a dependent WLM work unit

Non-swap service - OW26631

SWAP SERVICE (OW26631 R3

C function:

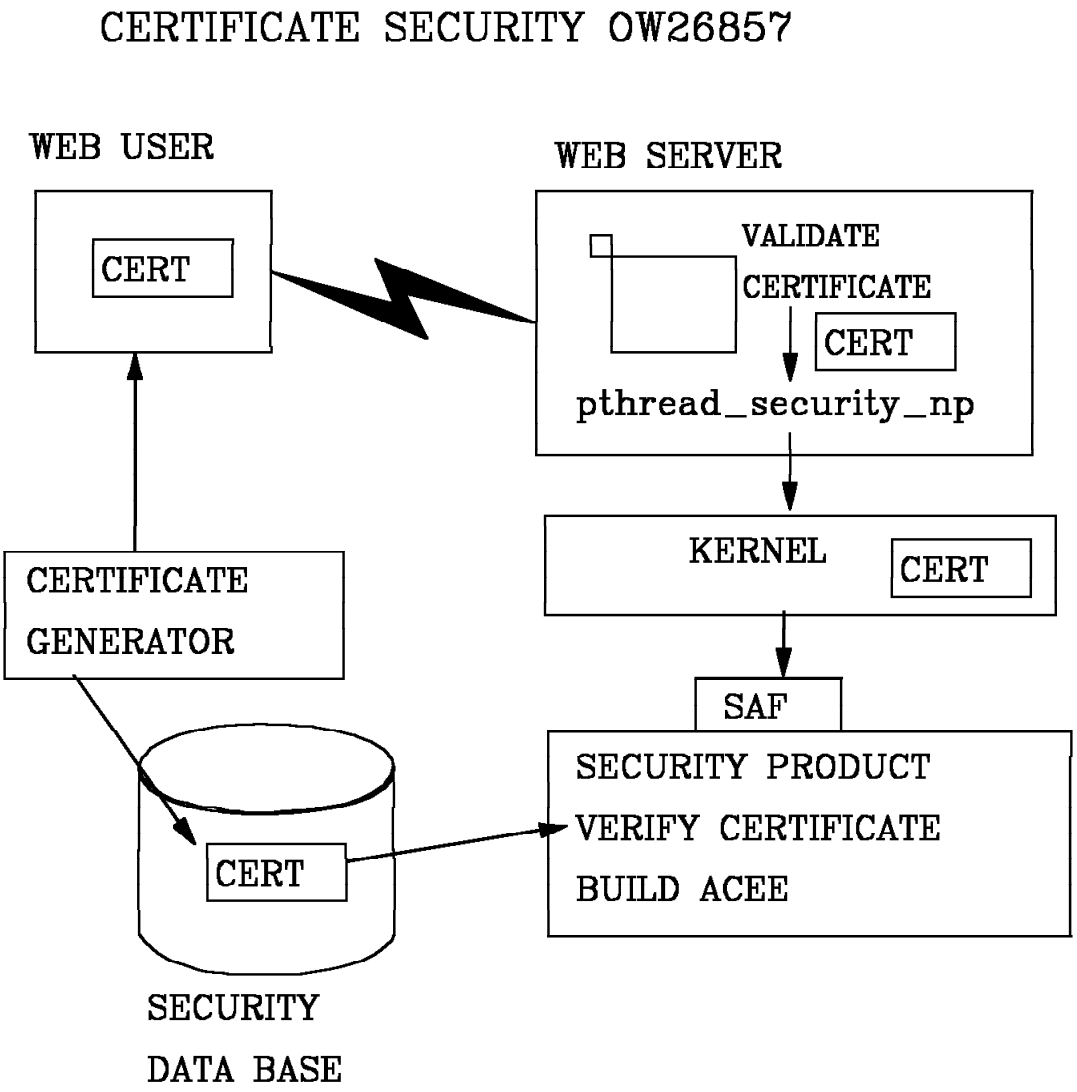
__mlockall(NONSWAP)



- Also option to do SYSEVENT OKSWAP
- Doesn't require APF authorization

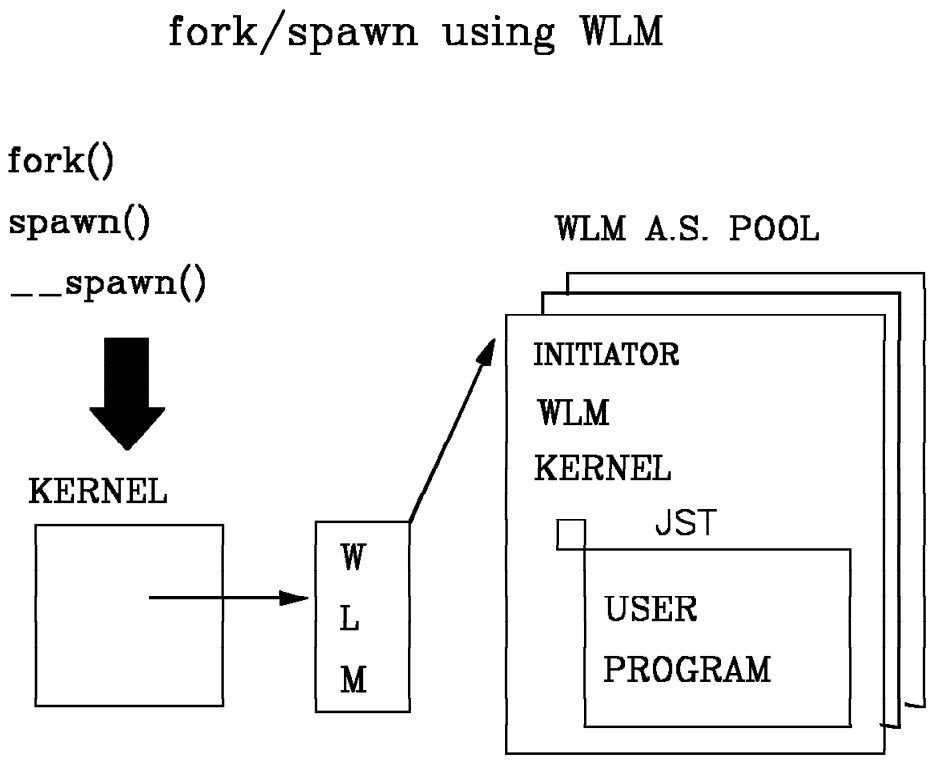
SWAP

Certificates - OW26857



CERT

WLM spaces for fork/spawn



- APPC is no longer used by fork and spawn
- WLM grows and trims the address space pool as needed based on goals. No user limits to manage.

WLMFORK

Extended Attributes - R4

EXTENDED FILE ATTRIBUTES – R4

- PROGRAM CONTROL
- APF Authorized
- Program does not run in a shared address space (for use with spawn to override _BPX_SHAREAS=YES)

extattr – shell command to modify extended attributes

ls –E – Displays extended attributes

ISHELL support to set and display as well

Ability to set extended attributes controlled by FACILITY class profiles:

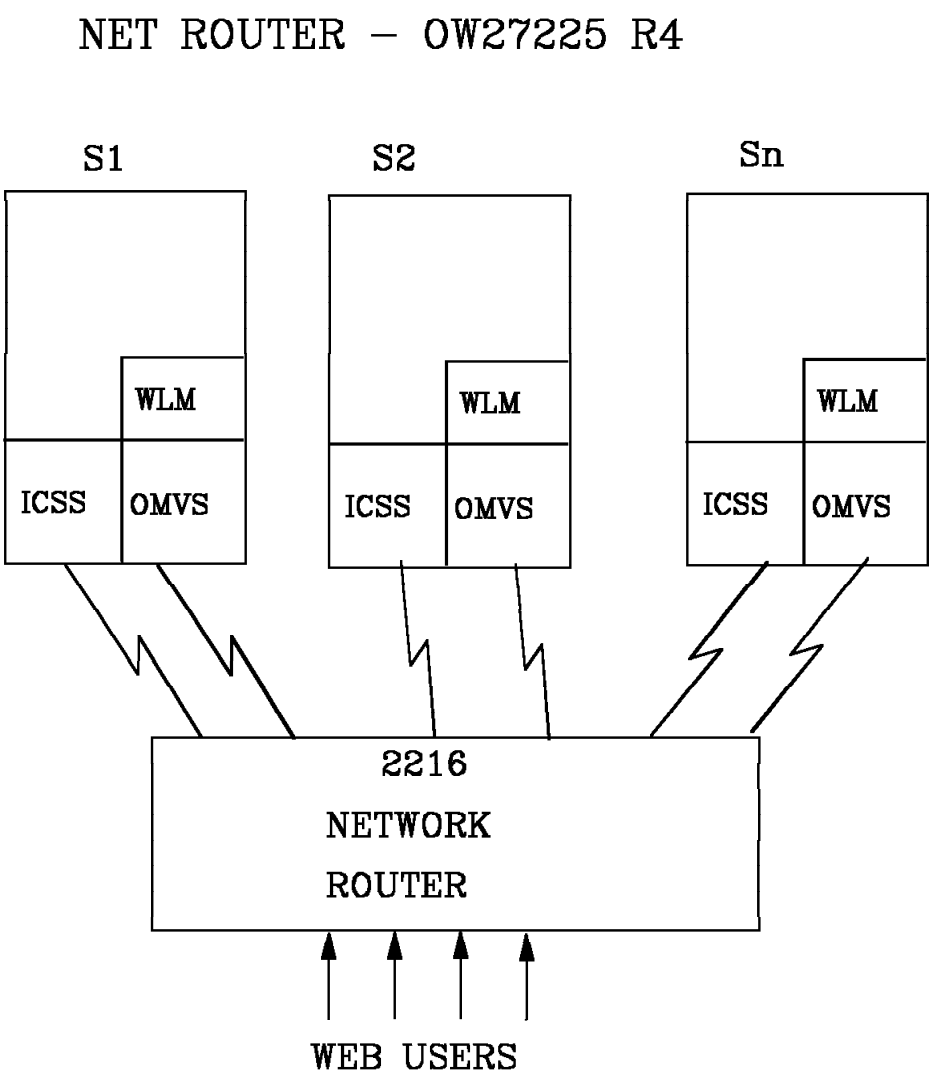
BPX.FILEATTR.PROGCTL

BPX.FILEATTR.APF

Updates to a file turn off PROGCTL and APF attributes.

APFHFS

Network Router 2216 - R4

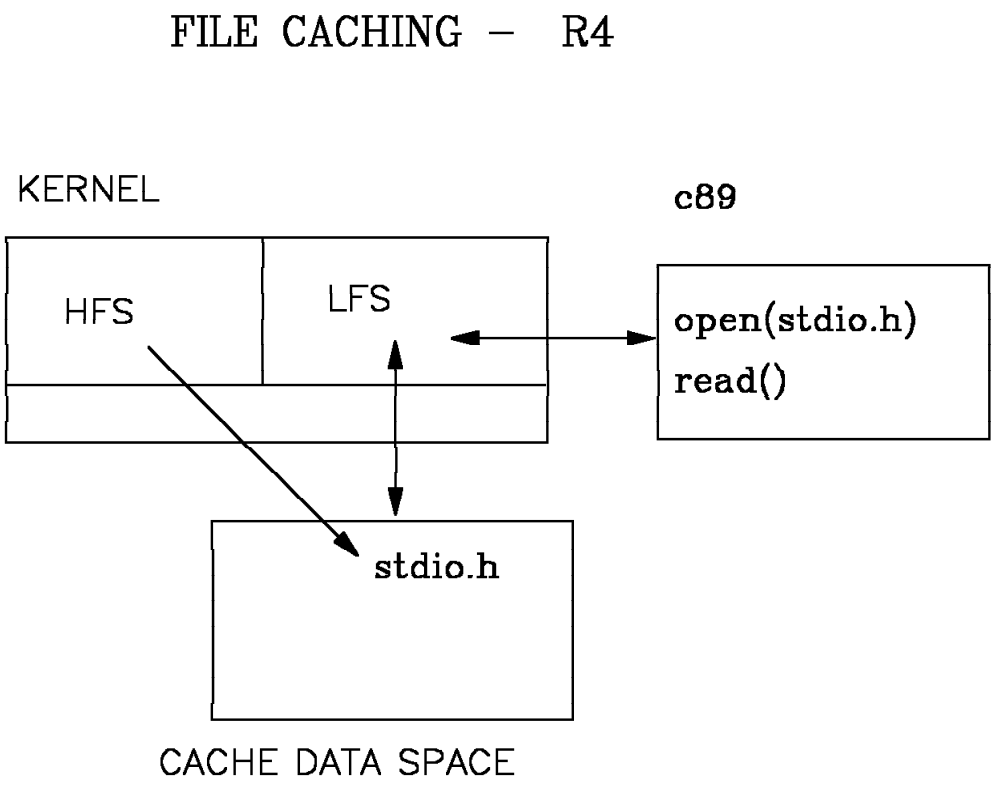


Router gets workload info from OE /WLM.

Router chooses best system for new requests.

NETROUT

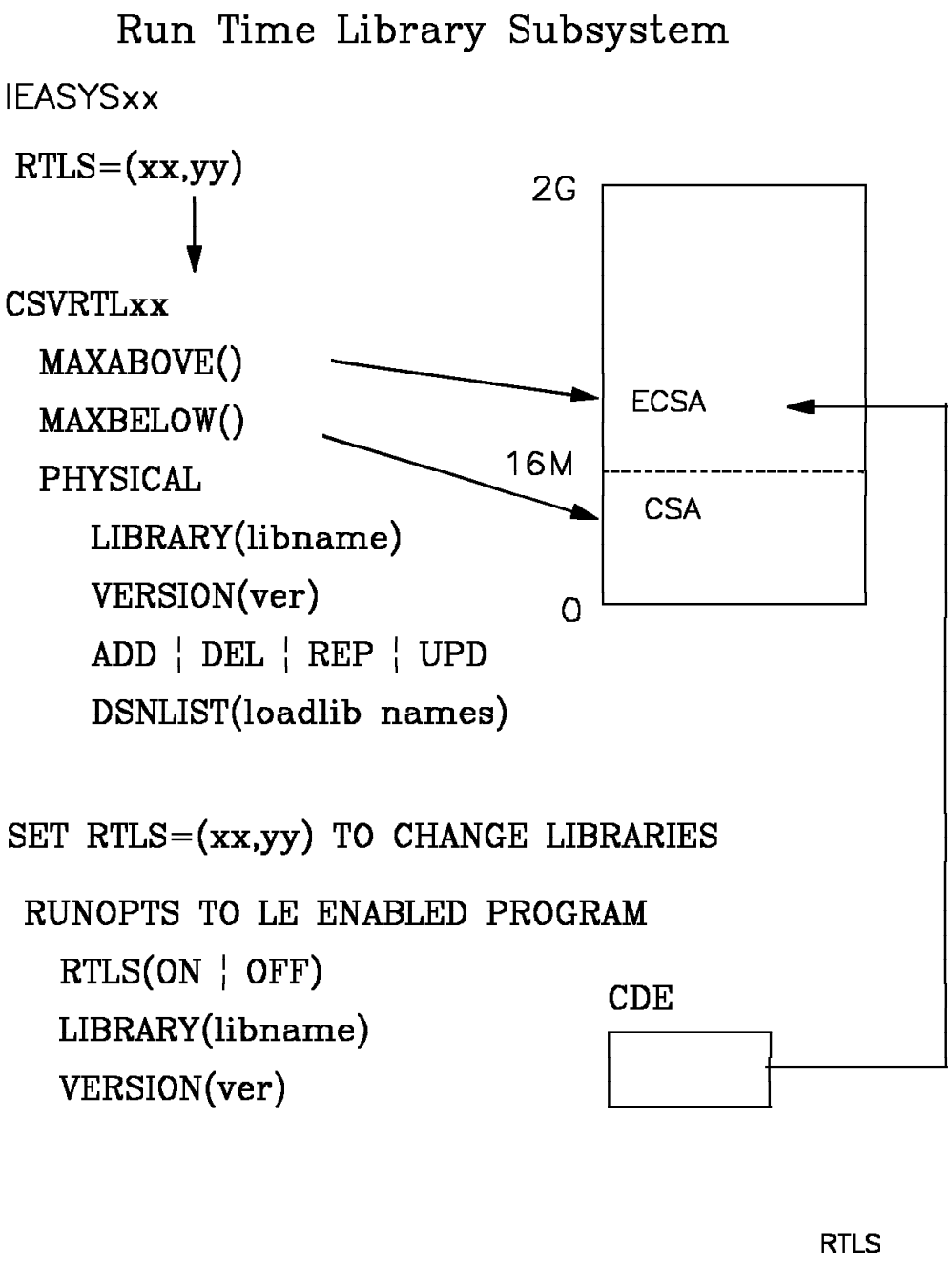
File Caching - R4



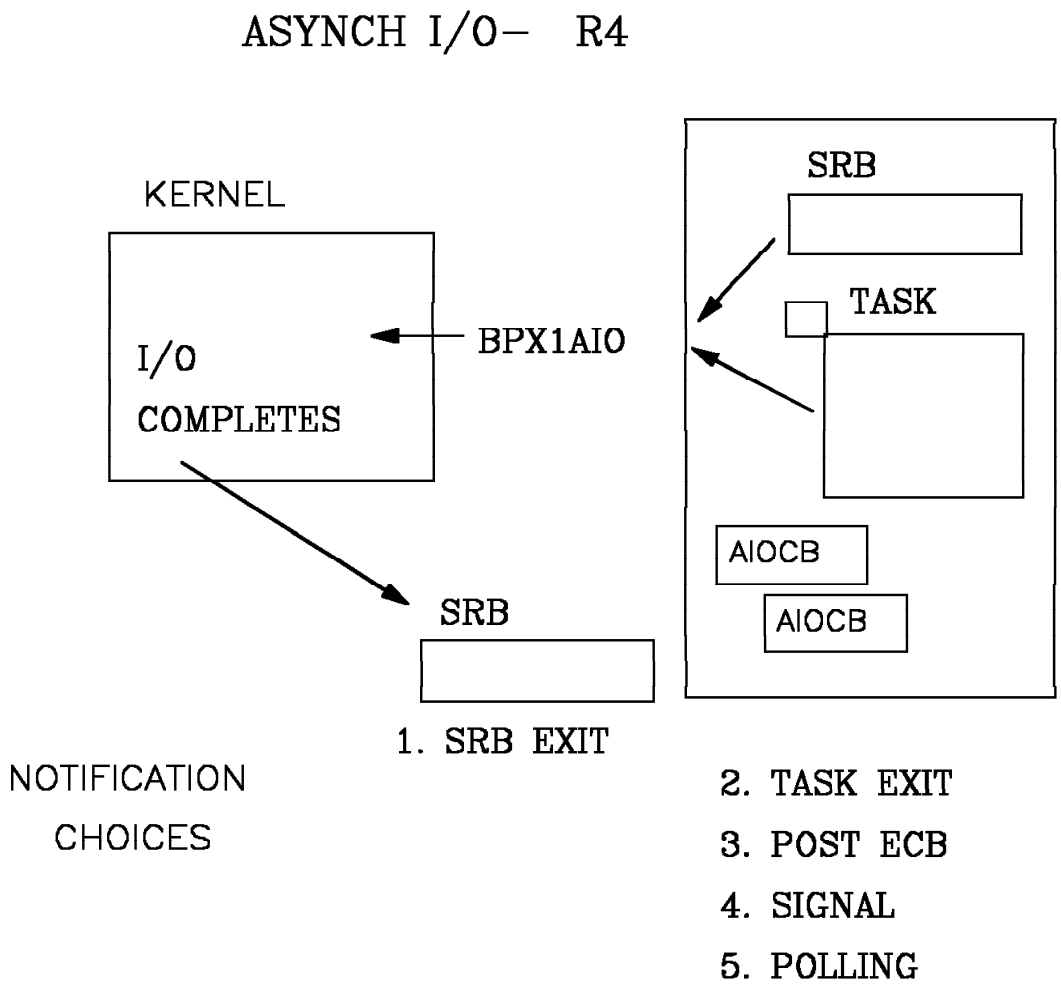
- Updates to files in the cache flush it from the cache
- Meant for small frequently used read only files like header files or frequent web pages
- shell command to add, delete, or refresh cache

FILECACH

RunTime Library Subsystem (RTLS) - R4



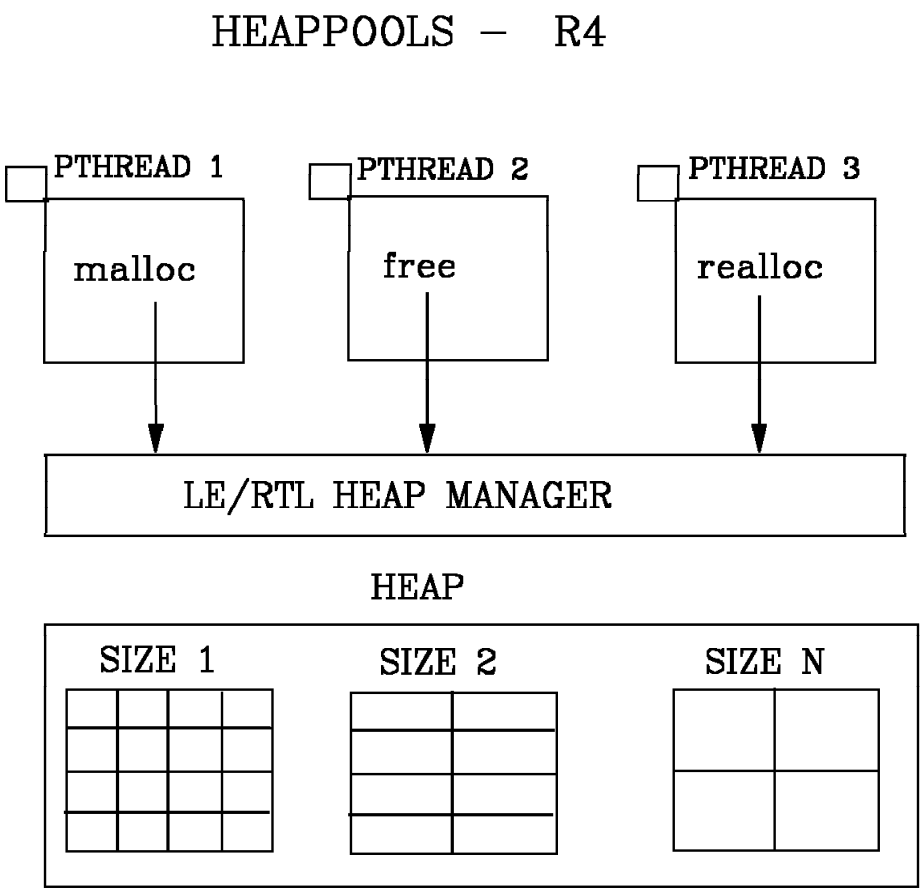
Asynchronous I/O (sockets) - R4



- Use instead of select on group of sockets
- C interface coming with UNIX 98 interface

AIO

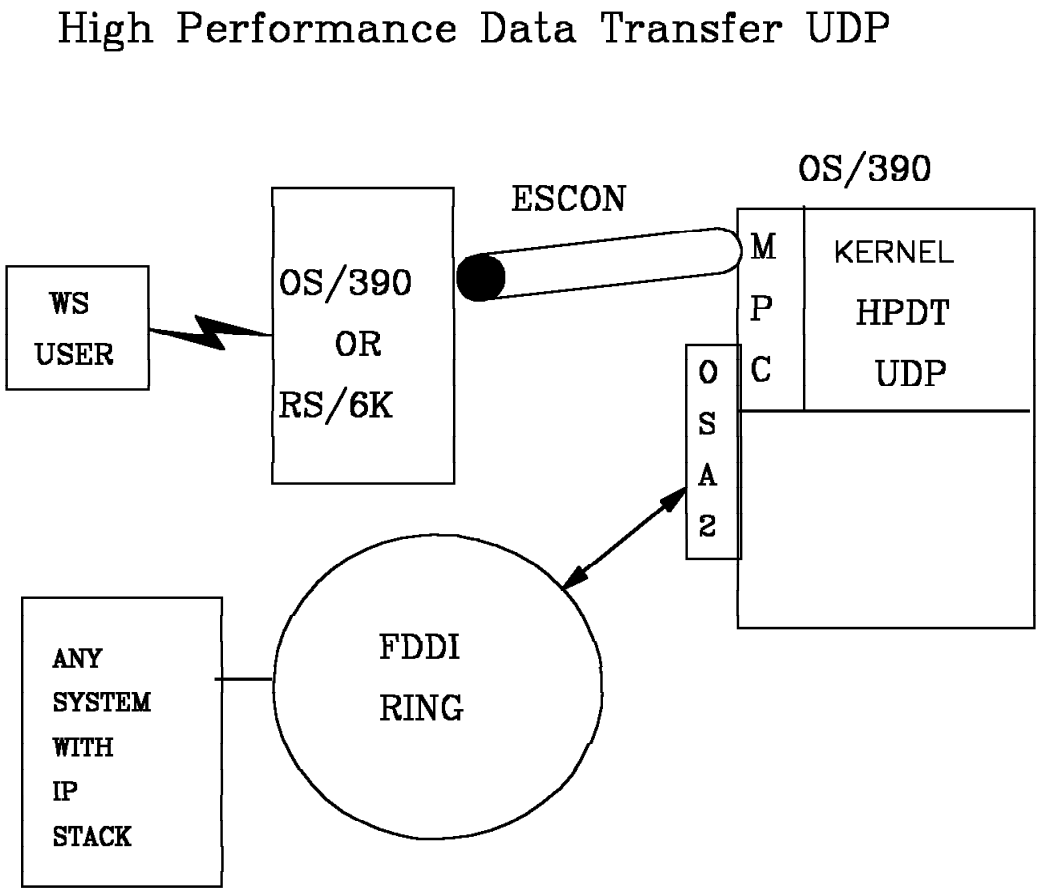
New heap manager option - HEAPPOOLS



- Heap managed as a set of *pools*
- Avoids serialization (uses CDS)
- 4 times faster and no scaling effects
- HEAPPOOLS run time option

HEAPPOOL

High Performance Data Transfer UDP



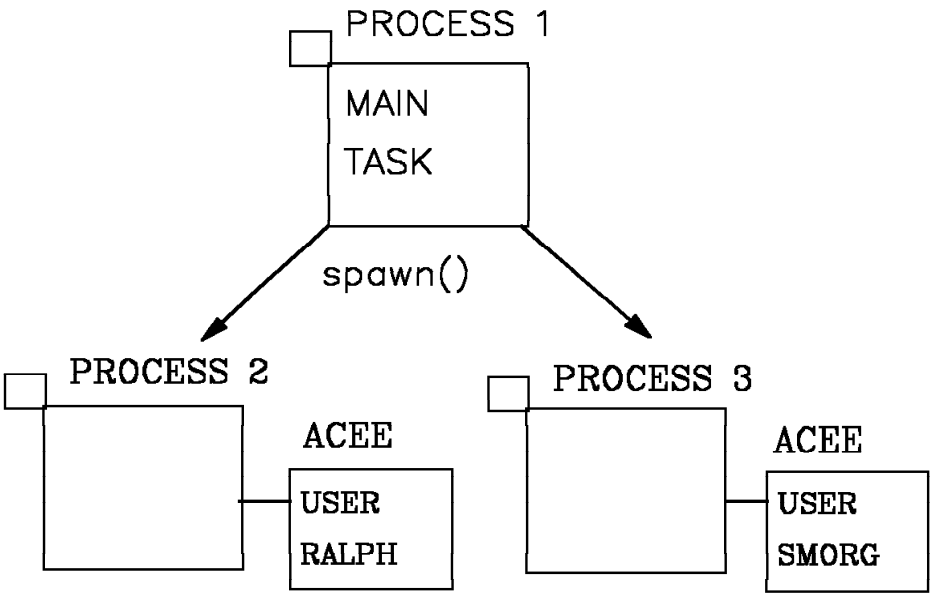
- Not constrained to single TSP/IP stack
- Primarily lock free
- Minimum pathlength
- Optimized blocking algorithm

HPDTUDP

Other R4 Enhancements

- iconv() performance
- LE Heap damage locator
- LE CEEDUMP enhancements
- IPCS VERBX LEDATA formatter
- LE replace ESPIE with ESTAE
- SVC Dump exit for SYSOMVS CTRACE
- IP Address Resolution Cache
- DBX support for debug of authorized processes
- REXX enhancements

MultiProc/MultiUser (OW27256 R4)



- __login to create task level ACEE for local process
- exec preserves task level ACEE
- force local spawn option to create process

MPMU

Future possibilities

- UNIX 98
- Access Control Lists (ACLs)
- 64 bit
- sysplex
- Shared memory reduction in usage of real memory
- Performance (we'll never be finished)

Forums

IBM TALKLINK -> OPENMVS CFORUM

Georgetown mailing list

```
send note to LISTSERV@LISTSERV.Georgetown.edu
body of note
     SUB MVS OE <your name>

Archieve for MVS OE Georgetown mailing list
  http://www.stortek.com/cgi bin/oe arc.cgi
```

Websites

http://www.s390.ibm.com/products/oe/index.html

```
OpenEdition Home Page (text only version)
 UNIX95 graphic   information about XPG4 branding
 Feature stories (changing)
   * Download KornShell ¢93
   * What¢s in Release 3 ?
   * Performance/tuning targets
   * Try our Setup Checker
 PORTING graphic   takes you to porting page
 Regular Features:
    * About OE         general background information
                       about OpenEdition servic
    * Tools & Toys     free, downloadable packages
    * Big Lists        Lists of suppported C functions and shell commands
    * Vendor Apps      List of commercial applications ported to OE
    * FAQs             User questions, sorted into 6 categories:
                       the HFS, the shell, porting, programming,
                       the compiler, system administration.
    * dbx              beta code, documentation, information
                       about bugs, apars, etc.
    * Public Forums    Info about public forums and OE mailing list
    * Links            links to related IBM products,
                       other companies, resources
    * OS/390 Elements      List of base and optional elements of OS/390
    * Site Map          Text Guide to the information on the OE web site
    * Porting           OE porting page (described on separate foil)
    * The Library       Access softcopy OE books on web, list of OE books,
                        recommended redbooks
```

OE Web Site

Tools and Toys Page http://www.s390.ibm.com/oe/bpxa1toy.html

Free, downloadable packages. New ones added regularly.

Ported Tools: ACE, banner, cpost, dtet2, gnumake, gzip, ksh93, m4,
ncftp, perl, ping, Samba, uuencode, uudecode, wall

Toys (homegrown tools):

cploadmod	(copies load modules between MVS and HFS)
dirsize	(shows amount of data in directory and its subdirectories)
getuids	(shows info about OE users and groups from security DB)
ifind	(searches file system for linked files)
libascii	(ascii interface layer for commonly used C RTL functions)
oesvp	(OE installation and setup verification program)
omvstape	(read/write to tapes during a shell session)
osendmail	(send mail to remote shell users)
perr	(outputs message text given a hex error or reason code)
pschart	(lists processes, showing parent child relationships)
qftp	(ftp capabilities for shell user)
readmvs	(copies MVS data set to stdout)
writemvs	(copies stdin to MVS data set)
stat	(displays stat() info about specified file)
startd	(starts daemons that operate synchronously)
submit	(shell command that submits JCL)
uusrestore	(restores trailing blanks in uuencoded file)

Function packages:

shcmd	(rexx function that runs a shell command)
rexxfunc	(package of I/O functions and other functions)
bpxwdyn	(dynamic allocation/output interface for REXX and C)

OE Web Site ...

Porting Page http://www.s390.ibm.com/oe/bpxa1por.html

News Stories (examples)
 * An ASCII interface package for commonly used C RTL functions
 * MKS offers Samba for OS/390 at its ftp site
 * Rogue Wave ports tools.h++ product set
 * AT&T Ports UNIX Tools
 * A list of applications ported to OpenEdition
 * Lotus and IBM announce S/390 Server Platform will support
     thousands of Lotus users

The Porting Book (published on web, chapter by chapter)
 * Setting up to port: your OS/390 UNIX environment
 * Sizing the port
 * Database migration
 * Process management

Porting Tips
 * Porting pthreads
 * Portable header files
 * ASCII to EBCDIC conversion
 * OpenEdition supplied C functions: a list
 * Porting questions and answers
 * Compiler questions and answers

Porting services and resources

System Verification Program

The Setup Verification Program (downloadable from Tools & Toys page)

After you have
completed your OpenEdition setup and customization
(including the shell and utilities), you can run the setup verification
program (SVP). The SVP will:

* Verify that each user has a UID and OMVS segment defined, and each
  group has a GID

* Check for duplicate assignment of UIDs and GIDs.

* Verify that each user has access to and owns a home directory and has
  read, write, search access to it

* Check the permissions for several directories usually set up at
  installation

* Check that files in the /dev directory are defined correctly.
  Reconcile the number of pseudo ttys and
  file descriptor files with the BPXPRMxx definitions.

* Verify that the shell will run.

* Verify that the OMVS command will run.

* Check customization for utilities. The program checks:

    files that have been copied from /samples to /etc
    terminfo files
    settings for some environment variables
    ability to compile and run a program

and performs various other checks.
If it detects a problem, the SVP warns you about it and, if you request,
corrects it.
The SVP can take up to one half hour to complete; time depends on your
system

Conclusion

Our goal in OS/390 UNIX is to have the most ITY's:

- Reliability
- Availability
- Serviceability
- Portability
- Scalability
- Possibilities