

**What an MVS Programmer  
Needs to Know About OS/390  
UNIX**

Kershaw Mehta  
kershaw@us.ibm.com

Session Number: 2877

Permission is given to SHARE to publish this presentation in its proceedings. IBM retains the right to distribute copies of this presentation to whomever it chooses.

August 20, 1998

# Table of Contents

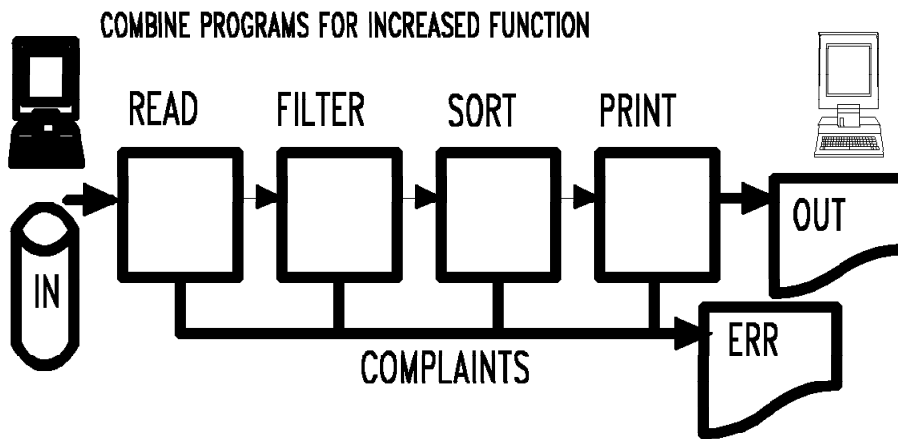
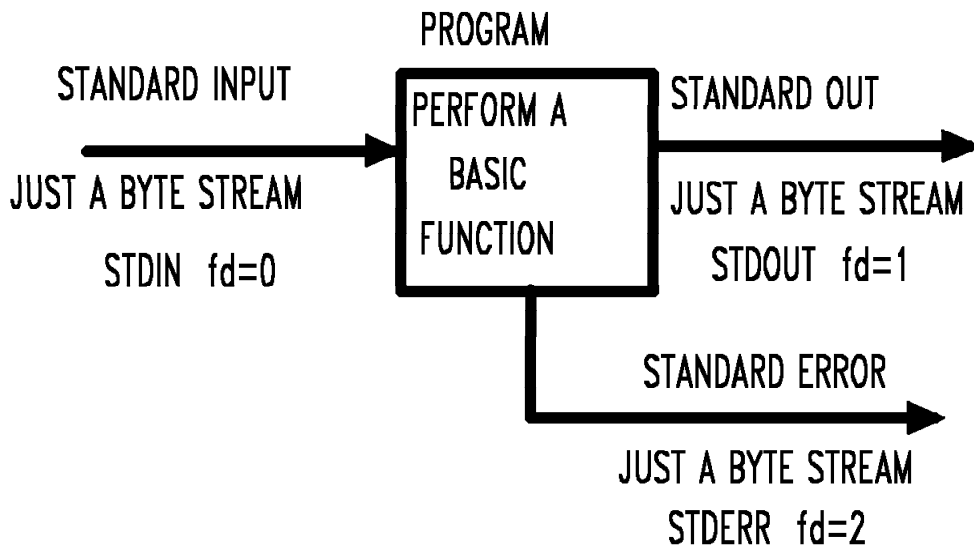
---

Getting to know the UNIX side	1
BPXPRMxx	9
Jobnames	10
Region Size	11
CPU Time Limits	12
Job Wait Time Limit	13
HSM Considerations for HFS	14
Operator commands	15
F BPXOINIT	17
SLIP for HFS (R3)	18
Taking Dumps	19
IPCS - for system problems	20
Facility Class Profiles	22
Program Control	23
RACF Setup	24
Default UID/GID - OW27564	25
Mixed Case Alias Userids	26
CTRACE	27
ISPF Dialog	28
ISHELL	29
Forums	30
Websites	31
OE Web Site	32
System Verification Program	34

# Getting to know the UNIX side

---

## \*IX SHELL PHILOSOPHY

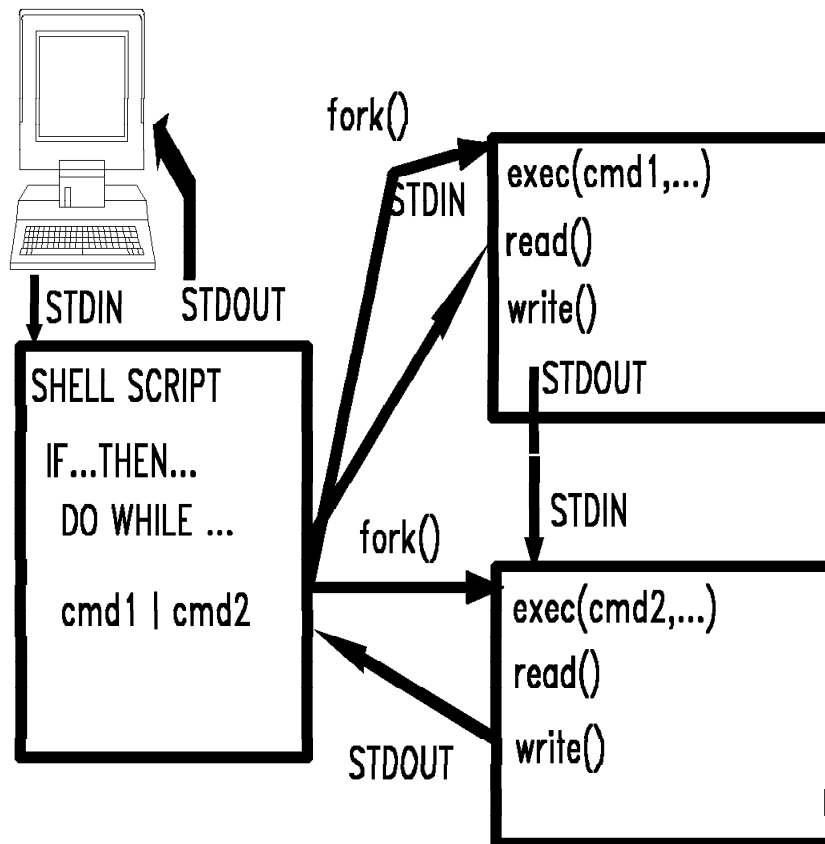


SHELLPH

# Getting to know the UNIX side ...

---

## SHELL & UTILITIES



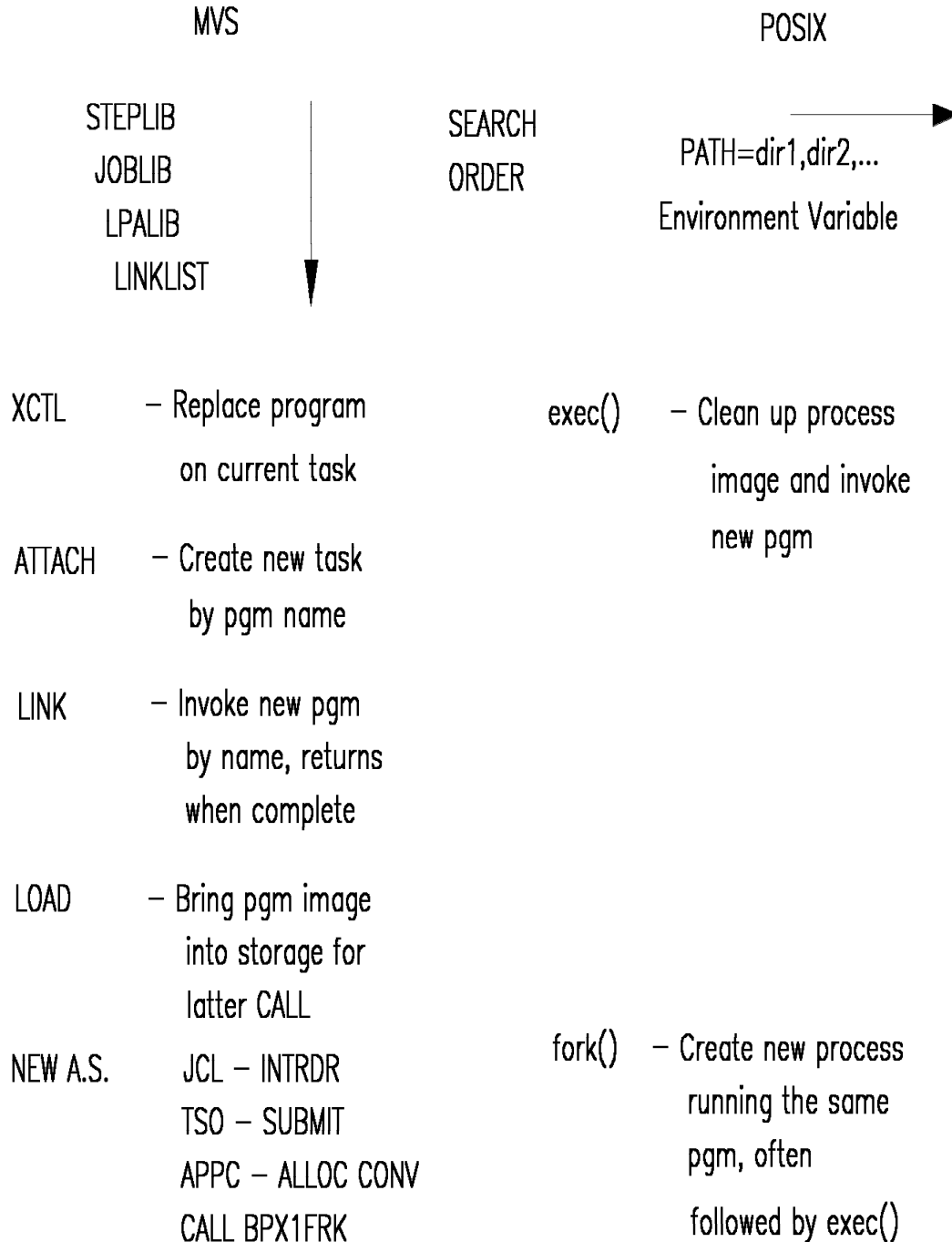
- flexible interpretive language
- file scan facilities
- character manipulation
- command piping

SHELL1

# Getting to know the UNIX side ...

---

## PROGRAM MANAGEMENT

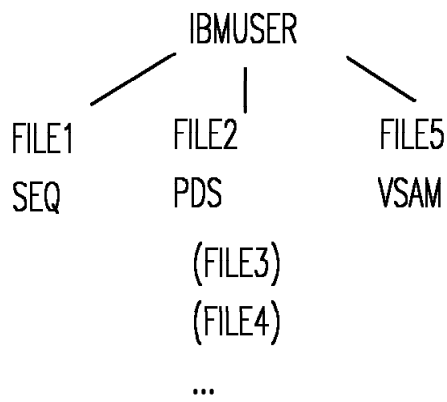
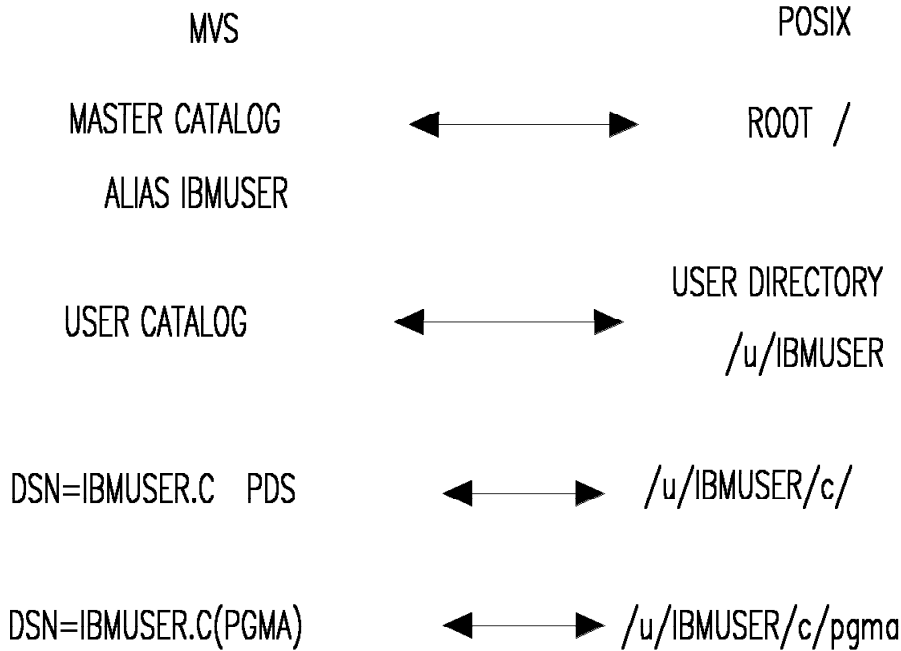


MVPCSV

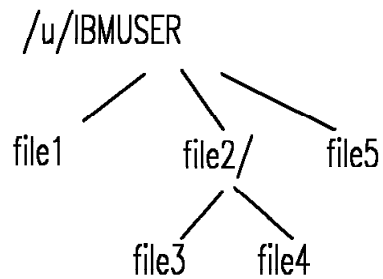
# Getting to know the UNIX side ...

---

## FILE SYSTEMS



RECFM, BLKSIZE,  
TYPE OF DATASET



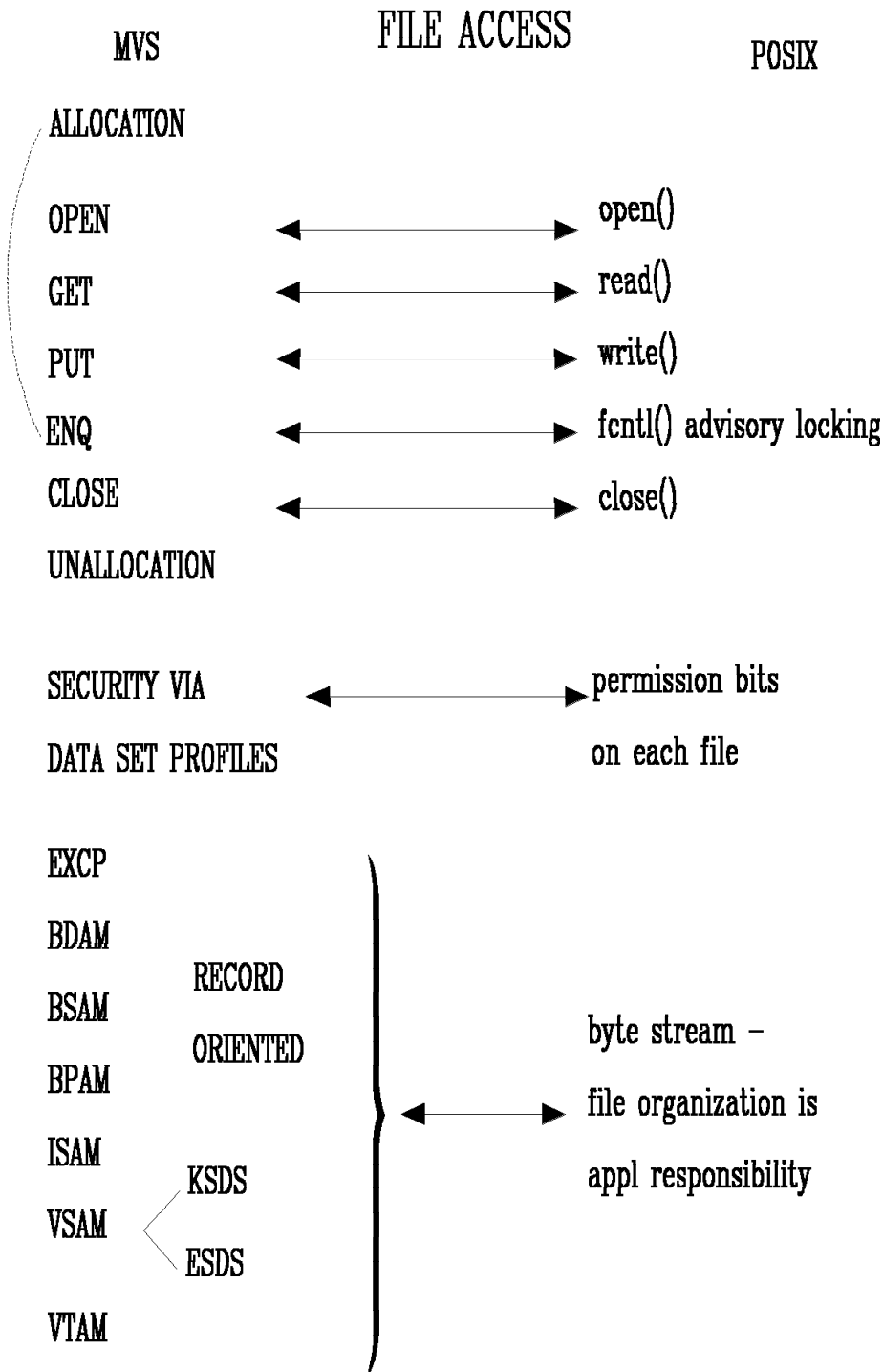
Organization provided  
by the application

Bytes  $\mathcal{R}$  Us

MVPFILE

# Getting to know the UNIX side ...

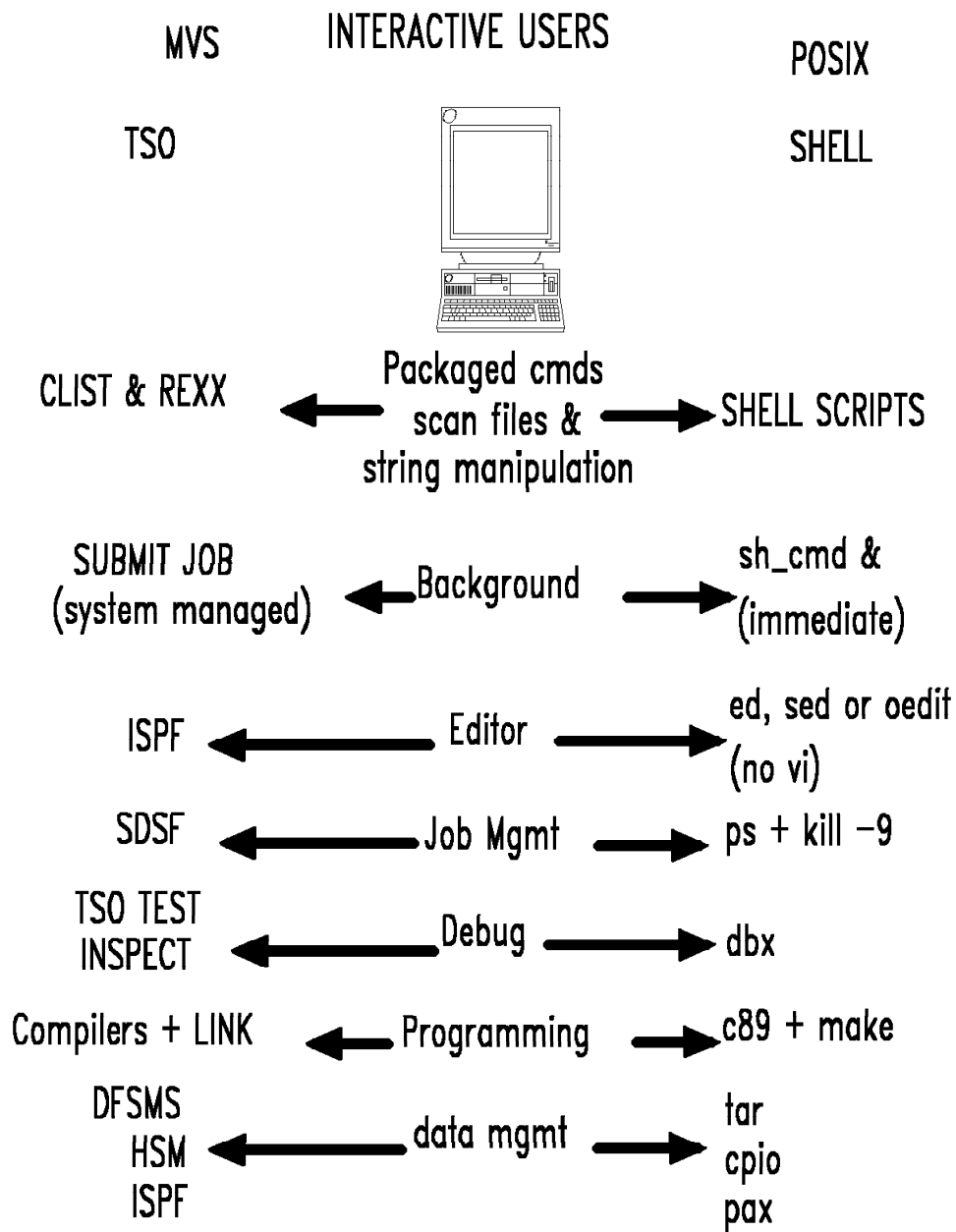
---



MVFILE2

# Getting to know the UNIX side ...

---



MVPIU

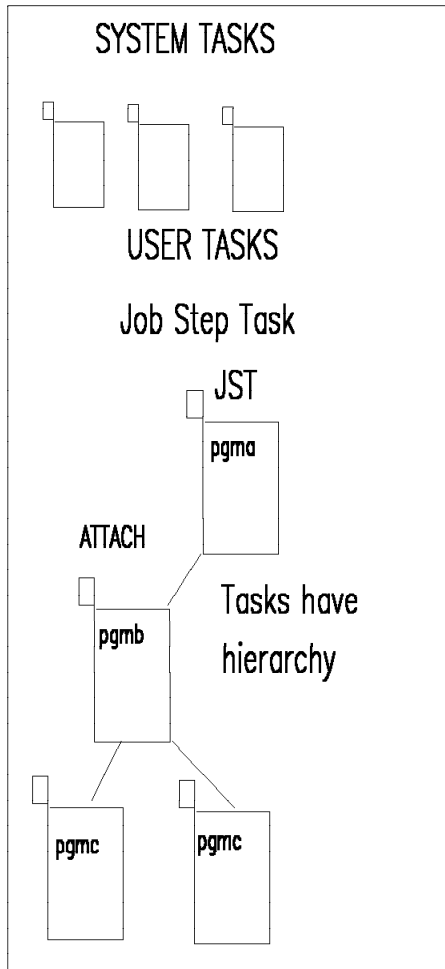


# Getting to know the UNIX side ...

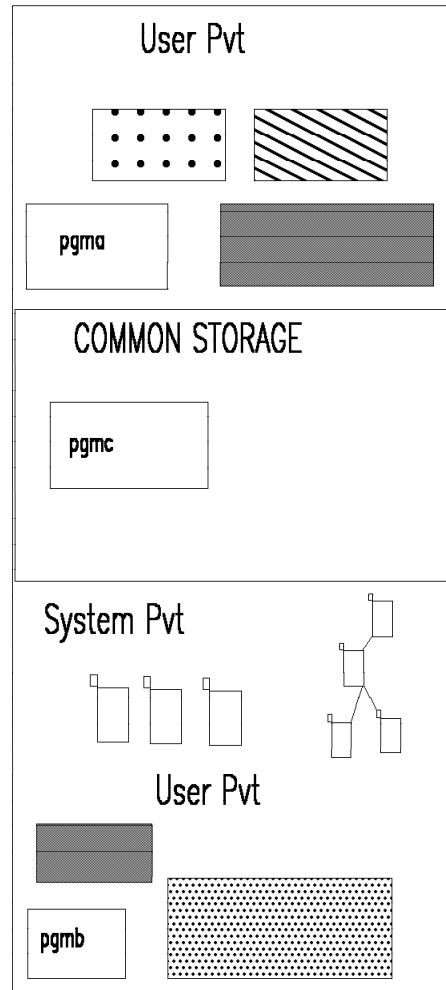
---

## MVS TASKS

### USER ADDRESS SPACE



### MEMORY MAP

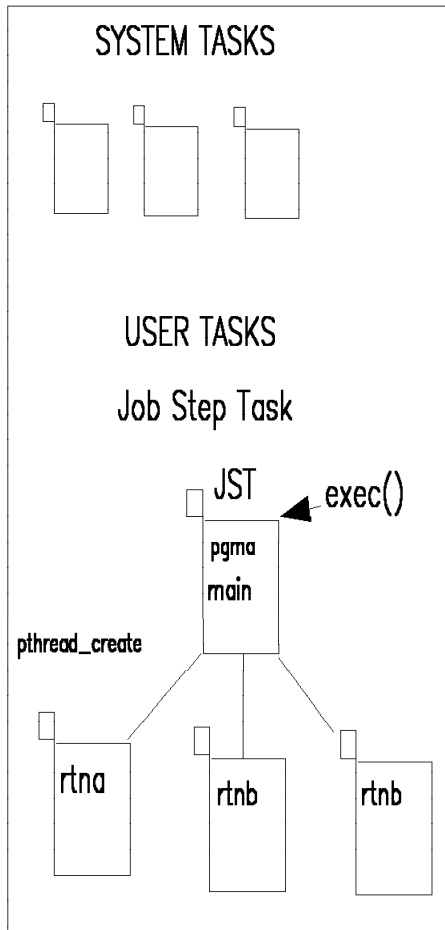


- Tasks may be independent or cooperating.
- Task storage is private and/or shared between tasks
- MVS tasks used where POSIX uses fork
- New task is ATTACHed by program name

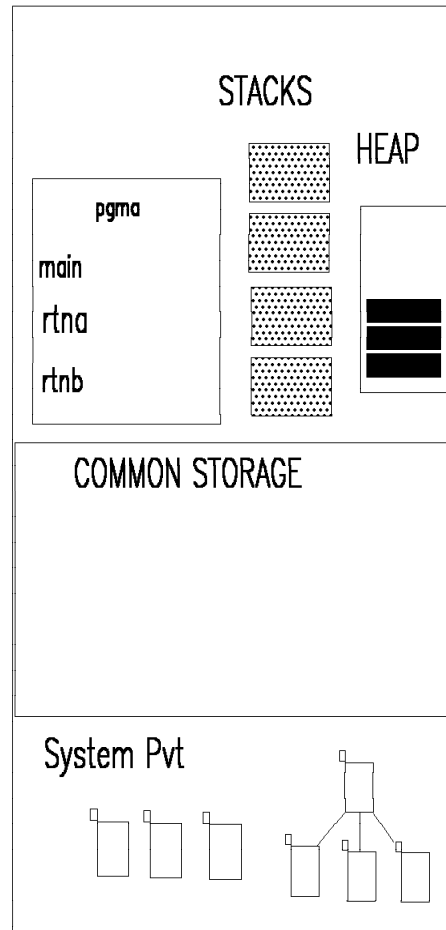
# Getting to know the UNIX side ...

## POSIX PTHREADS ON OpenMVS

### USER ADDRESS SPACE



### MEMORY MAP



- User creates task with `pthread_create` by address
- All code comes from same load module
- Threads have no hierarchy
- Threads execute asynchronously and concurrently
- Typically only one task create by `exec()` = JST

POSTASK

# BPXPRMxx

---

- Template shipped in SYS1.SAMPLIB(BPXPRMXX)
- MAXPROCSYS=nnn - max processes in system  
No limit for max forked address spaces. WLM controlled.
- MAXPROCUSER - Max processes for a single UID.
- Other various limits: threads, shmem, semaphores, files
- Define file system types (HFS, sockets)
- Define ROOT
- Define mount points
- Define automount policy

# Jobnames

---

RULES for child process on fork and spawn:

- If parent jobname is 8 char, child gets same jobname
- If parent jobname less than 8 char, child gets jobname+1-9
- Jobname changed to userid on exec after a setuid
- Jobname changed to userid on spawn with userid
- Jobname can be set on spawn/exec with `_BPX_JOBNAME=jobname`

Example of jobnames on rlogin request:

- INETD running with jobname INETD
- Gets request and forks creating INETD1
- Setuid to userid rlogind
- Exec rlogind changing the jobname to RLOGIND
- Verify userid=RALPH/password and setuid
- Exec rlogind2 changing the jobname to RALPH
- Fork issued and child gets jobname to RALPH1

## Region Size

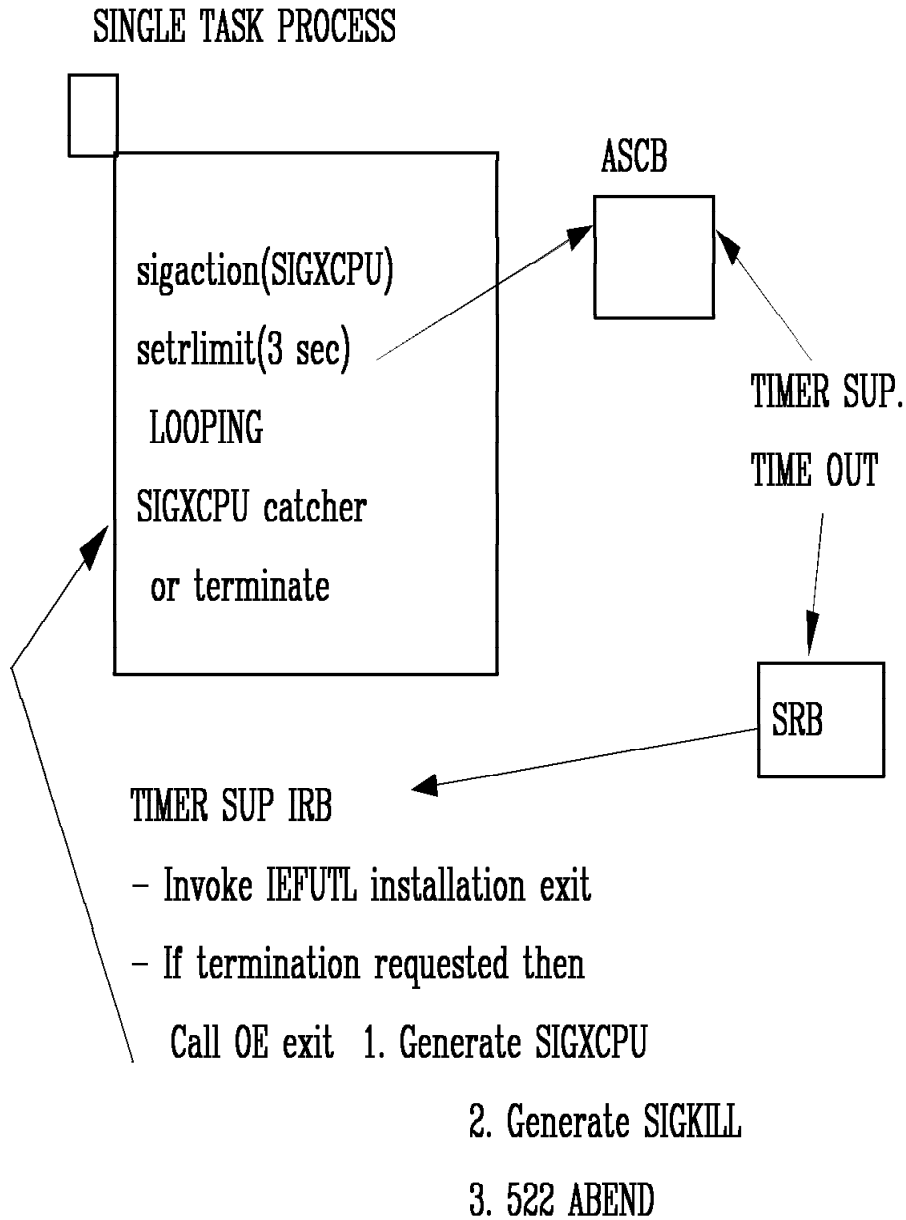
---

- When dubbed, we leave REGION size alone
- On fork, we propagate REGION from parent to child
- On exec following setuid or spawn with userid, REGION is set to MAXASSIZE.
- setrlimit() can be used to change your region size
- If IEFUSI changes the region size, we try to honor it.

# CPU Time Limits

---

## SETRLIMIT for CPU TIME



SETRL2

## Job Wait Time Limit

---

- JWT value specified in SMFPRMxx
- Signal enabled waits:
  - Cause kernel to raise wait limit to 7 years
  - Allows parent to wait for child
  - Reset on wake up.
- The shell supports TMOU

# HSM Considerations for HFS

---

- HSM performs HFS Quiesce prior to dump
- Quiesce only works on locally mounted HFS
- Global ENQ done if can't quiesce - fails if mounted elsewhere
- Therefore, you can only have HSM dump HFS's that are:
  - Mounted on same system as HSM
  - Not mounted on any system in sysplex
- Won't be fixed till we have sysplex HFS support



# Operator commands

---

- DISPLAY OMVS - file systems and processes
- D GRS,C - for latch contention
- D TRACE,CT,COMP=SYSOMVS
- SET OMVS=(xx,yy, ...)
- SETOMVS keyword=value
  - Supports most limit values in BPXPRMxx
  - Does not currently support filesystem statements

# Operator commands ...

---

## D GRS,C

```
ISG020I 19.20.18 GRS STATUS 806
NO ENQ RESOURCE CONTENTION EXISTS
LATCH SET NAME:  SYS.BPXFLIT.FILESYS.LATCH.SET
CREATOR JOBNAME: OMVS      CREATOR ASID: 0038
  LATCH NUMBER:  691
    REQUESTOR  ASID  EXC/SHR  OWN/WAIT
    KERSHAW    0047  EXCLUSIVE  OWN
    WPID013    002A  EXCLUSIVE  WAIT
  LATCH NUMBER:  879
    REQUESTOR  ASID  EXC/SHR  OWN/WAIT
    RALPH      005F  EXCLUSIVE  OWN
    SIGGD531   0055  EXCLUSIVE  WAIT
  LATCH NUMBER:  2819
    REQUESTOR  ASID  EXC/SHR  OWN/WAIT
    SMORG      004C  EXCLUSIVE  OWN
    SIGGD534   003A  EXCLUSIVE  WAIT
```

## D OMVS,A=ALL

```
BPX0001I 19.22.27 DISPLAY OMVS 809
OMVS      ACTIVE
USER      JOBNAME  ASID      PID        PPID  STATE  START
OMVSKERN  BPX0INIT  0022      1          0     1WI    15.47.22
KERSHAW   KERSHAW    0047     1245197    1     1KI    17.20.53
OMWPID    OMWPID     0047     5177362    1245197 1KIT   17.20.58
OMSIGGD   SIGGD531   0055     14483477   1     1KI    17.59.00
OMSIGGD   SIGGD534   003A     11272218   1     1KI    18.34.24
OMSIGGD   SIGGD531   004C     9371676    11272218 1KIT   18.34.29
OMSIGGD   SIGGD531   005F     786499     14483477 1KIT   17.59.13
```

# C KERSHAW,ASID=47

## **F BPXOINIT**

---

### **F BPXOINIT,TERM,PID=pppppppp**

Send a SIGTERM to specified PID. Like kill from shell.

### **F BPXOINIT,TERM,PID=pppppppp.ttttttttttttt**

Send a SIGTERM to specified thread. Designed to knock out a thread in a server like SAP or web.

### **F BPXOINIT,SHUTDOWN=FORKINIT**

Tell WLM to take down the fork initiators.

# SLIP for HFS (R3)

---

## SLIP SUPPORT FOR HFS – R3

SLIP SET,IF,PVTMOD='/bin/sh',...

SLIP SET,C=0C4,PVTMOD='inetd',...

- Allows up to 8 characters which are matched at end of file name that was loaded.
- Supports already loaded and future loads
- Supports exec, spawn, DLLs and HFSLOAD.
- If sticky, then use existing SLIP keywords

NEW DUMP EXIT – OS/390 R4

Dumps OMVS CTRACE when OMVS is not the one that took the dump. Also covers DUMP and SLIP commands.

SLIP

# Taking Dumps

---

**IF YOU GET A HANG**

**AND NEED A DUMP**

**DUMP COMM=(Hung user with OMVS)**

**R xx,SDATA=(CSA,SQA,RGN,TRT,GRSQ),CONT**

**R xx,ASID=(E,47,92),CONT**

**R xx,DSPNAME=('OMVS'.SYSZBPX2),END**

Where E is the kernel, 47 could be the TSO space and 92 a forked addresss space.

Only need SYSZBPX2 if you suspect file system problems.

# **IPCS - for system problems**

---

## **OMVSDATA subcommand**

- REPORTS - PROCESS, FILE SYSTEM, COMMUNICATIONS, STORAGE, IPC
- 3 levels of detail
- ASID and USERID filtering
- OMVSDATA PROCESS DETAIL ASID(X'47') very useful

## **CBSTAT for TCB and ASCB**

**>60 CB Models - internal**

# IPCS - for system problems ...

---

## OMVSDATA PROCESS DETAIL

```
Process ID: 00010002
Status: Active
Last exec() Program Name: No exec issued
ID Data:
    Userid: TC                      Asid: 0047
    Parent PID: 00000001          Ptrace Parent PID: 00000000
    Process Group ID: 00010002    Session ID: 00010002
    Real Uid: 22                  Real Gid: 500
    Effective Uid: 22             Effective Gid: 500
    Saved Set Uid: 22            Saved Set Gid: 500
    Foreground PGid: 00010002
    Process Group Member IDs: 00010002
    Session Member IDs: 00010002
Signal Data (Process Level):
    All Signal Data In Default State
File System Data:
    Current Working Directory (CWD) Name (at time of last chdir()):
    /
    CWDs File Serial Number: 00000000          CWDs Device Number: 00000001
    Number of open files for this process:      0 Token: 00400070
Thread Data: Active Threads
    Thread ID: 02DEA0B0 00000000
    Tcb Address: 009FC378
    Program Name:
    ASID(X'001E') 02D00EF4. TPSIG999+03E4 IN EXTENDED PRIVATE
    Thread Attributes:
        undetached
        heavy
```

## Facility Class Profiles

---

- BPX.DAEMON - controls changing identity (setuid)
- BPX.DEBUG - Allow DBX of APF authorized program
- BPX.FILEATTR.APF - Allow file owner to mark file APF auth
- BPX.FILEATTR.PROGCTL - Allow file owner to mark file program controlled
- BPX.SERVER - controls changing identity (pthread\_security\_np)
- BPX.SMF - Allows use of smf\_record service
- BPX.STOR.SWAP - Allows you to make address space non-swappable  
Beware of impact on storage reconfiguration
- BPX.SUPERUSER - allow setuid to uid=0
- BPX.WLMSEVER - allow use of WLM interfaces



# Program Control

---

When a UNIX program is performing identity changes, forcing the program control requirement, makes it harder for a user to attack daemon code to increase their authority.

BPX.DAEMON and BPX.SERVER trigger check for Program Control.

All programs in the address space must be program controlled before things like `setuid` or `pthread_security_np` will work.

# RACF Setup

---

```
ADDUSER SMORG DFLTGRP(ENGNP7) NAME(¢RALPH SMORG¢)  
      OMVS(UID(314) HOME(¢/u/smorg¢) PROGRAM(¢/bin/sh¢))  
      WORKATTR(WANAME(¢RALPH SMORG¢) WAACNT(12345678)  
      plus other parms
```

```
ALTGROUP SMORGRP OMVS(GID(7))
```

```
PERMIT BPX.SUPERUSER CLASS(FACILITY) ID(SMORG) ACCESS(READ)
```

```
PERMIT BPX.DAEMON CLASS(FACILITY) ID(SMORG) ACCESS(READ)
```

# Default UID/GID - OW27564

---

```
ADDGROUP OEDFLTG OMVS(GID(777777))  
  
ADDUSER OEDFLTU DFLTGRP(OEDFLTG) NAME(çOE DEFAULT USERç)  
        OMVS(UID(999999) HOME(ç/ç) PROGRAM(ç/bin/shç))  
  
RDEFINE FACILITY BPX.DEFAULT.USER APPLDATA(çOEDFLTU/OEDFLTGç)  
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.

# Mixed Case Alias Userids

---

## USERID ALIAS TABLE

BPXPRMxx – Identifies HFS file with table definitions

USERIDALIASTABLE('/hfs\_file\_name')

MYUSERID	MyUserid
K61XDLBC	ralph
MORSE	-.-.-.-
DASHING	-----
:GROUPS	
MYGROUP	MYGroup
WEIRDGRP	a&./\$--

- Degrades performance when alias table is defined.
- Alias names are any character but blank
- Performance improved in OS/390 R3
- Weird userids can break tools (periods and slashes)
- UNIX functions see alias, MVS sees real userid
- Updates to table are periodically picked up (oper cmd for immediate)
- Security product is unaware of alias

ALIAS

# CTRACE

---

- CTIBPX00 is default
- TR CT,size,C=SYSOMVS,PARM=CTIBPXxx
- D TRACE,C=SYSOMVS
- Supports asid filtering
- Supports userid filtering, borrows JOBNAME to do it.  
JOBNAME=(USER1,USER2,...)
- Topic filtes: PROCESS, SIGNALS, FILE, SYSCALL, ...
- Size 16K to 4M - default 64K

# ISPF Dialog

---

## EDITING AND BROWSING HFS FILES

```
----- EDIT - ENTRY PANEL -----  
Command ==>  
  
Directory    ==>  
  
File name    ==>  
  
Profile name ==>  
  
Initial macro ==>
```

10,'CMD(OBROWSE)'

20,'CMD(OEDIT)'

COPY

MOVE

REPLACE

CREATE

EDIT

} External Command

ISHELL contains many file management functions and systems management tools.

ISPF

# ISHELL

---

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)...		

# Forums

---

IBM TALKLINK -> OPENMVS CFORUM

Georgetown mailing list

send note to [LISTSERV@LISTSERV.Georgetown.edu](mailto:LISTSERV@LISTSERV.Georgetown.edu)  
body of note  
SUB MVS OE <your name>

Archive 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 supported 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/bpxaltoy.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/lpxalpor.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