

z/TPF EE V1.1  
z/TPFDF V1.1  
TPF Toolkit for WebSphere® Studio V3  
TPF Operations Server V1.2



IBM Software Group

## *TPF Users Group Spring 2006*

Coming soon to an image near you . . .

# z/TPF File System Enhancements

Name : Stephen Record  
Venue : Database / TPFDF Subcommittee

AIM Enterprise Platform Software  
IBM z/Transaction Processing Facility Enterprise Edition 1.1.0  
© IBM Corporation 2006

## Agenda

- panic dumps
- inode allocation
- utilities
- security
- tar
- shared memory
- /proc

## But first, some clues to our motivation . . .

- Requirements that would be addressed by these proposals
  - ▶ FITS 1228041627 - shared memory monitoring tool
  - ▶ FITS 1228042021 - online shared memory displays
  - ▶ FITS 1228042529 - shared memory house keeping
  - ▶ FITS 1228043124 - better shared memory interfaces
  - ▶ FITS 1228044924 - verification for shared memory unlinks
  - ▶ TUG D04005 - file system back-up and restore utility
  - ▶ TUG D04006 - support tar functionality on TPF
  - ▶ TUG D04007 - ZFILE login / logout
  - ▶ TUG D04008 - file system userid / password administration
  - ▶ Language neutral access to TPF primitives
- Requirements to be considered later on
  - ▶ TUG D04009 - file system quota limits and monitoring

## More informative panic dumps

- APAR PJ30836 on PUT 02
- Issue the system error in the module which detected the error
- Facilitate support and improve maintainability of the file system

## Improved TFS inode allocation

- APAR PJ30802 on PUT 02
- Balances the allocation load across all available #IZERO records
- Prevents spurious warnings to the operator about inode depletion

## Additional file system utilities - statement of direction

- APAR PJ30898 slated for a future PUT
- New ZFILE mount command option
  - ▶ increase the size of a file system (FFS or PFS only)
  - ▶ make additional records in the base record type available
- New ZFILE dspsys command
  - ▶ display all records from a individual file or an entire file system
  - ▶ display the contents of the record buffer
- New ZFILE view command
  - ▶ display allocated record addresses and record IDs for an individual file or an entire file system
- New ZCACH DUMP option
  - ▶ dump the contents of the specified logical record cache
- New ZAVFS BUILD command option
  - ▶ specify the number of fixed file records to use

## File system security support - statement of direction

- APAR PJ30915 slated for a future PUT
- Control who can manipulate files in the file system
- Control who can execute specific operator commands
- Based on POSIX file access permissions
- ZOVFS
  - ▶ for administrator to manage file system users and groups
  - ▶ initialize, enable, or disable file system security support
  - ▶ create, delete, modify, or display user and group IDs
- ZPVFS
  - ▶ for users to gain access to the file system
  - ▶ login or logout, change password
  - ▶ display currently logged in user
- User exit ufve.c would facilitate encryption of passwords
- TOS support for password obfuscation would be provided

## ZFILE tar command - statement of direction

- Possible approach to satisfying the backup/restore requirements
- New ZFILE tar command (may be open source)
  - ▶ standard UNIX command to manipulate tape archives
  - ▶ support headers in either EBCDIC or ASCII
    - use ZFILE dd command to convert text files as needed
- New ZFILE pax command (may be open source)
  - ▶ POSIX command to provide portable archive interchange (x)
  - ▶ support EBCDIC headers only
- New ZFILE dd command options
  - ▶ simplify conversion of character set and line end sequence for files being moved between z/TPF and Linux or UNIX systems
- New ZFILE df command
  - ▶ standard UNIX command to display disk usage per file system
- New ZFILE du command (may be open source)
  - ▶ standard UNIX command to display disk usage per file



## Shared memory extensions - statement of direction

- Possible approach to satisfying the shared memory requirements
- New shmget() option
  - ▶ support use of 64-bit system heap for shared memory
- New ZFILE shmm command to manage shared memory
  - ▶ display shared memory values and limits
  - ▶ change shared memory (soft) allocation limits
    - number of shared memory frames (31-bit and 64-bit)
    - number of shared memory segments
  - ▶ unlink shared memory descriptors
  - ▶ remove shared memory segments
    - free system resources
- New user exit ushmm.c would allow mediation
  - ▶ called whenever a soft allocation limit would be exceeded
  - ▶ called whenever shmat() is issued

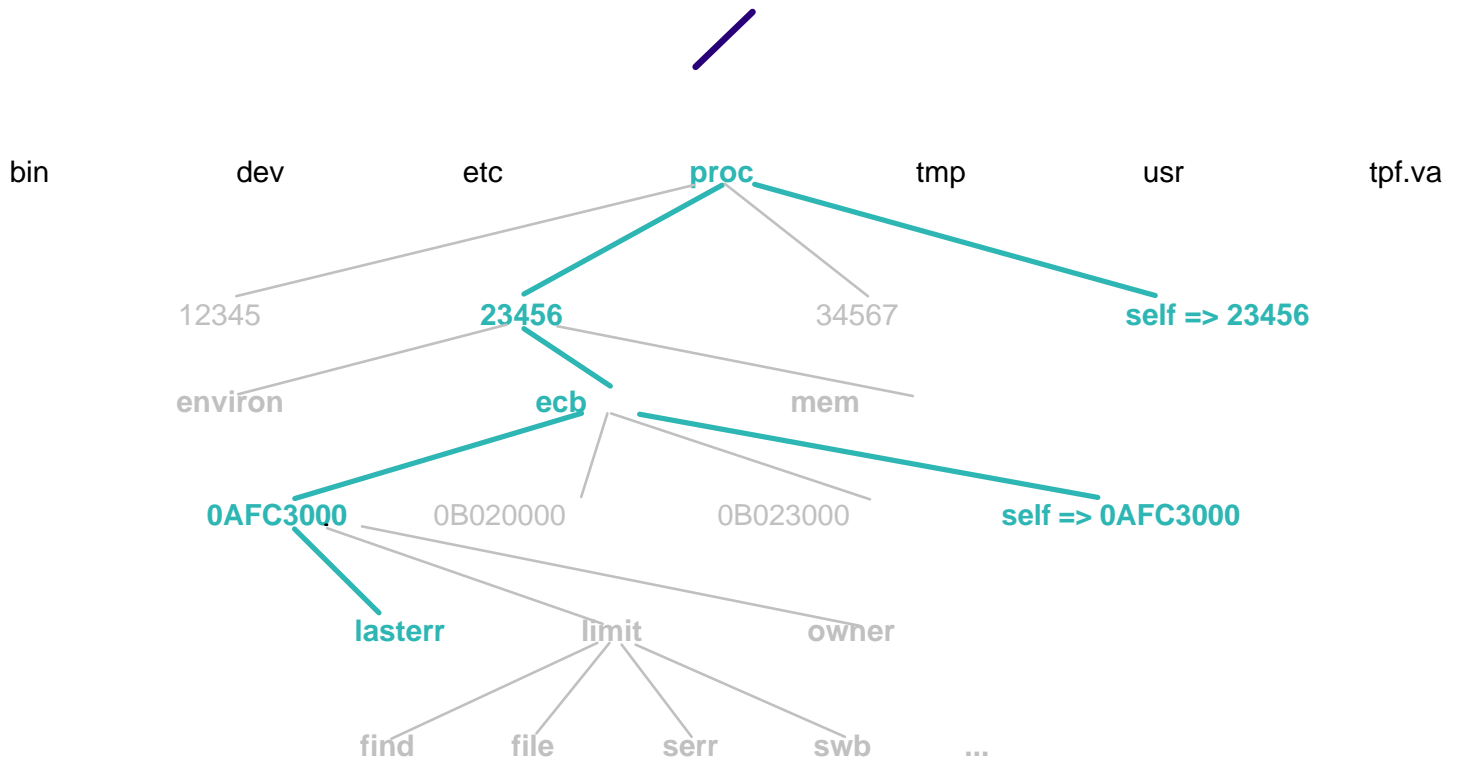
## Process pseudo-file system - statement of direction

- Possible means of providing language neutral access to z/TPF or POSIX primitives such as:
  - ▶ ECB owner information and resource limits
  - ▶ process environment and memory
  - ▶ TPF globals and events
- Implement a structure within which to provide interfaces to system services and data structures through the file system API
  - ▶ modeled on a long-standing UNIX design, also in Linux
  - ▶ exploit the virtual file system architecture
  - ▶ satisfy a requirement for access from languages like Java
  - ▶ provide easy extensibility for further development by both IBM and customers

## Process pseudo-file system - statement of direction (*continued*)

- Conventionally mounted on (and known familiarly as) /proc
- Top level subdirectories would represent active processes
  - ▶ the subdirectory name is the process ID
  - ▶ "self" is a symbolic link to the current process
- Under each process, subdirectories representing process level constructs would be added
  - ▶ the infrastructure would include the "ecb" subdirectory, under which the threads of the process are represented by their SVAs
  - ▶ under each SVA, the infrastructure would provide the "lasterr" file for diagnostic purposes
- Non-process-specific primitives (e.g. globals and events) could be represented in a companion pseudo-file system called /sys
  - ▶ direction taken in Linux 2.6 kernel

## Possible /proc hierarchy - infrastructure and extensions



## Trademarks

IBM, WebSphere, z/TPF, and z/TPFDF are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

### Notes

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.