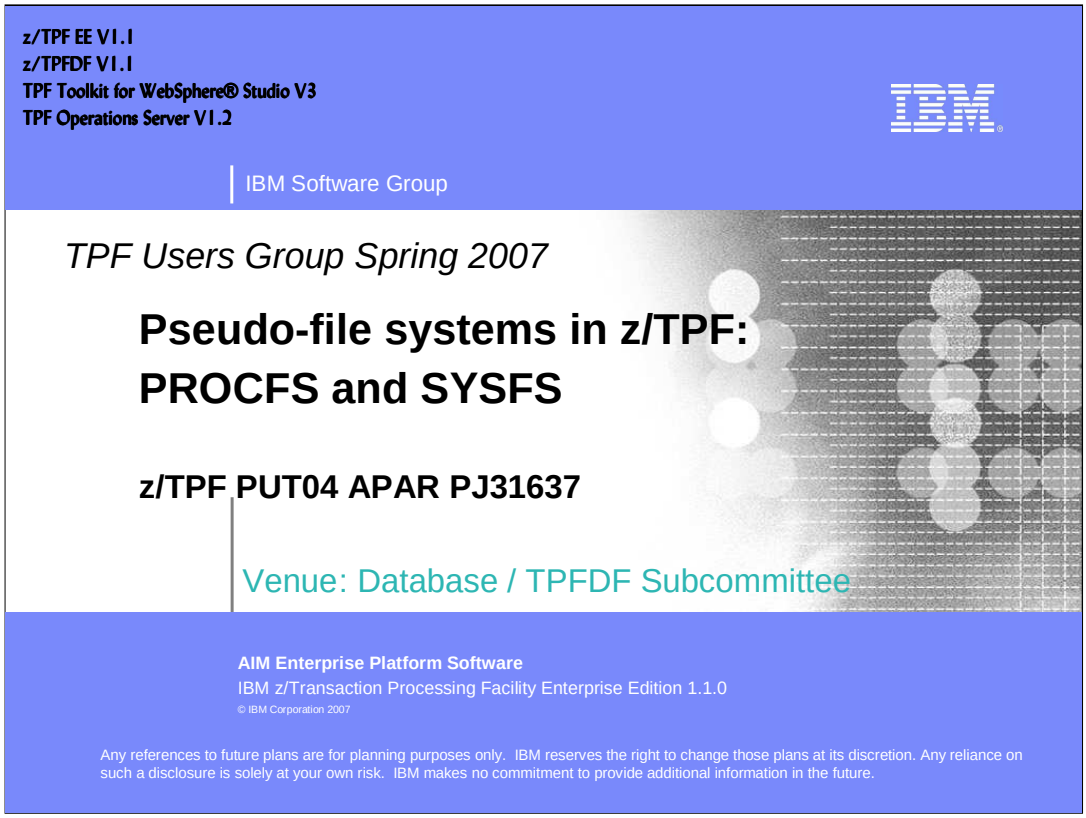
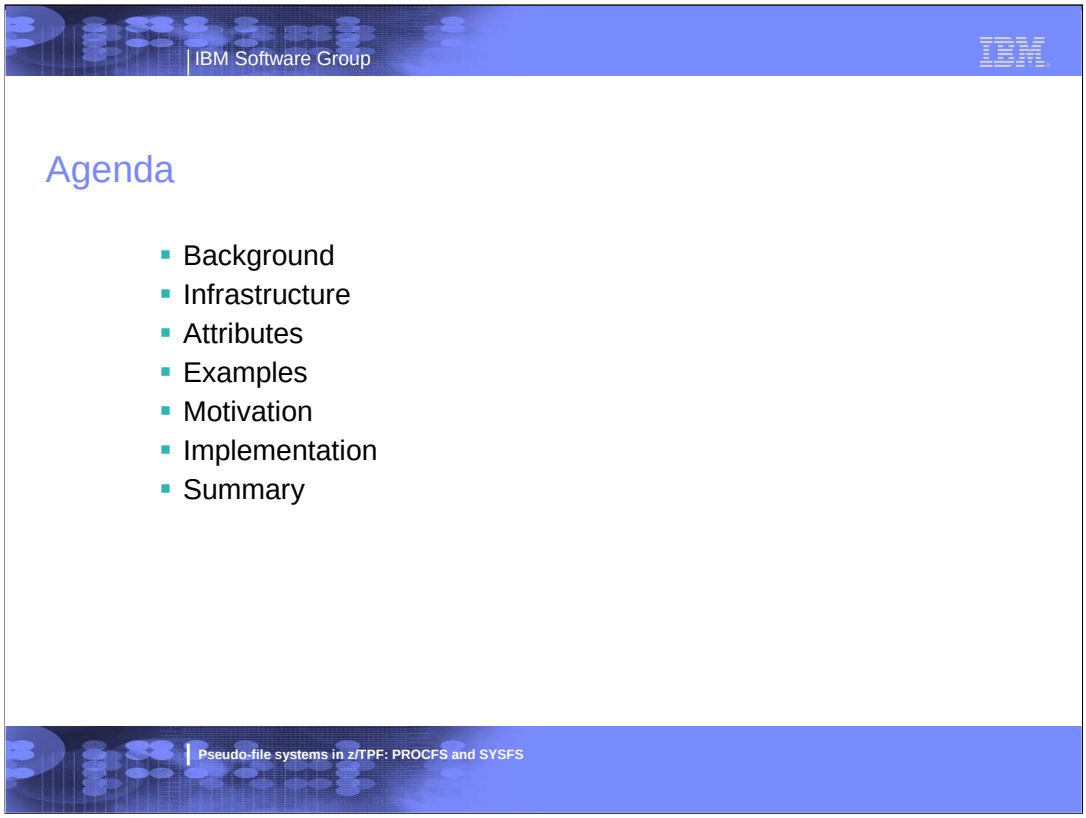


This presentation describes the pseudo-file systems PROCFS and SYSFS.

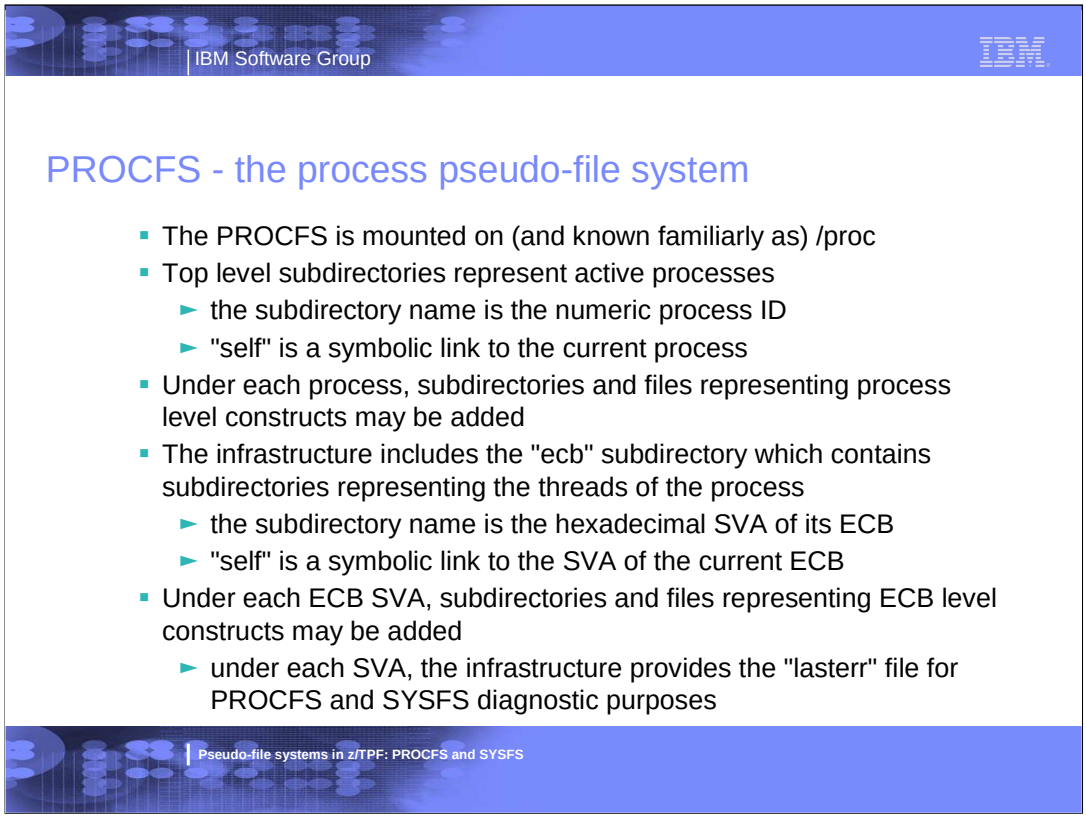


This presentation will describe PROCFS and SYSFS background information, infrastructure, attributes, examples, motivation, implementation, and conclude with a summary.

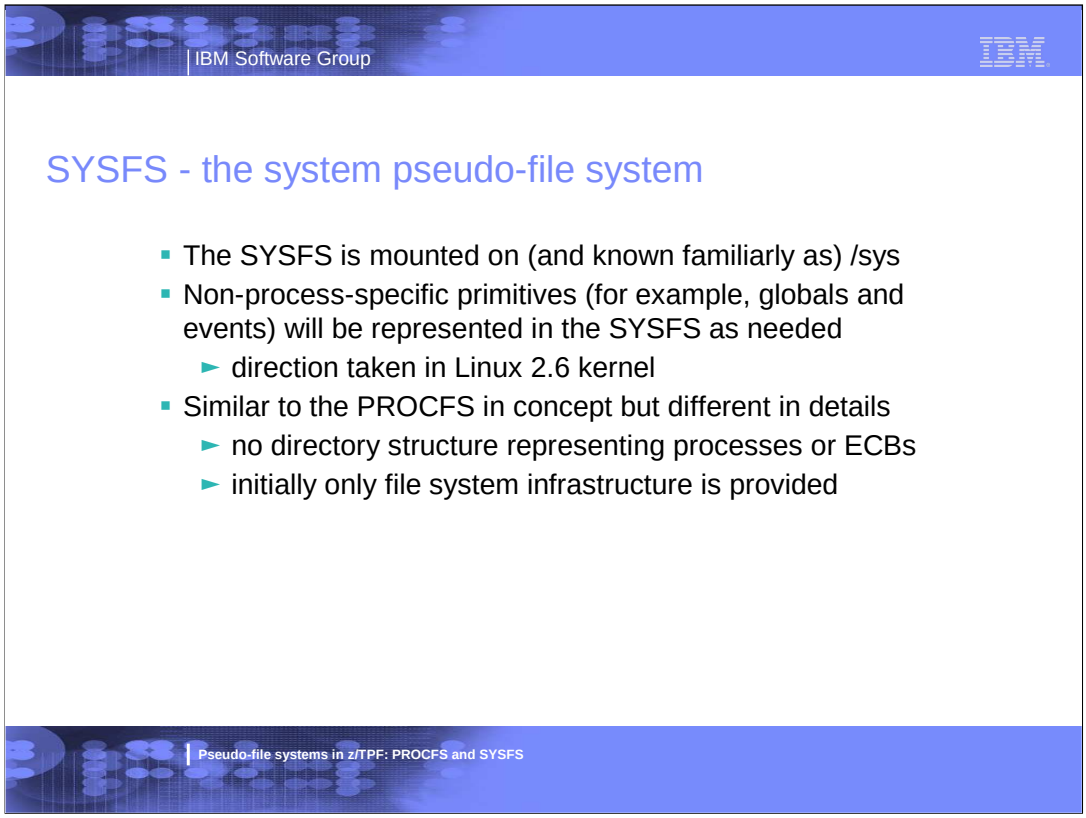
What is a pseudo-file system?

- Easier to say what it is not than to specify what it is....
- A Virtual File System (VFS) is an abstraction layer on top of a set of mountable "concrete" or "physical" file systems.
- The VFS allows programs to access the physical file systems in a uniform way (via a conventional file system API, in particular POSIX) without regard to the specific characteristics of their implementations.
- A pseudo-file system is a mountable file-system which has no conventional file store (for example, a dedicated pool of disk blocks) associated with it.
- The entities in a pseudo-file system (that is, regular files, directories, links) are instead instantiated by blocks of program code which simulate file system operations in some appropriate manner.
- The prototypical use of pseudo-files is to provide access to operating system kernel variables and data areas through the file system APIs.

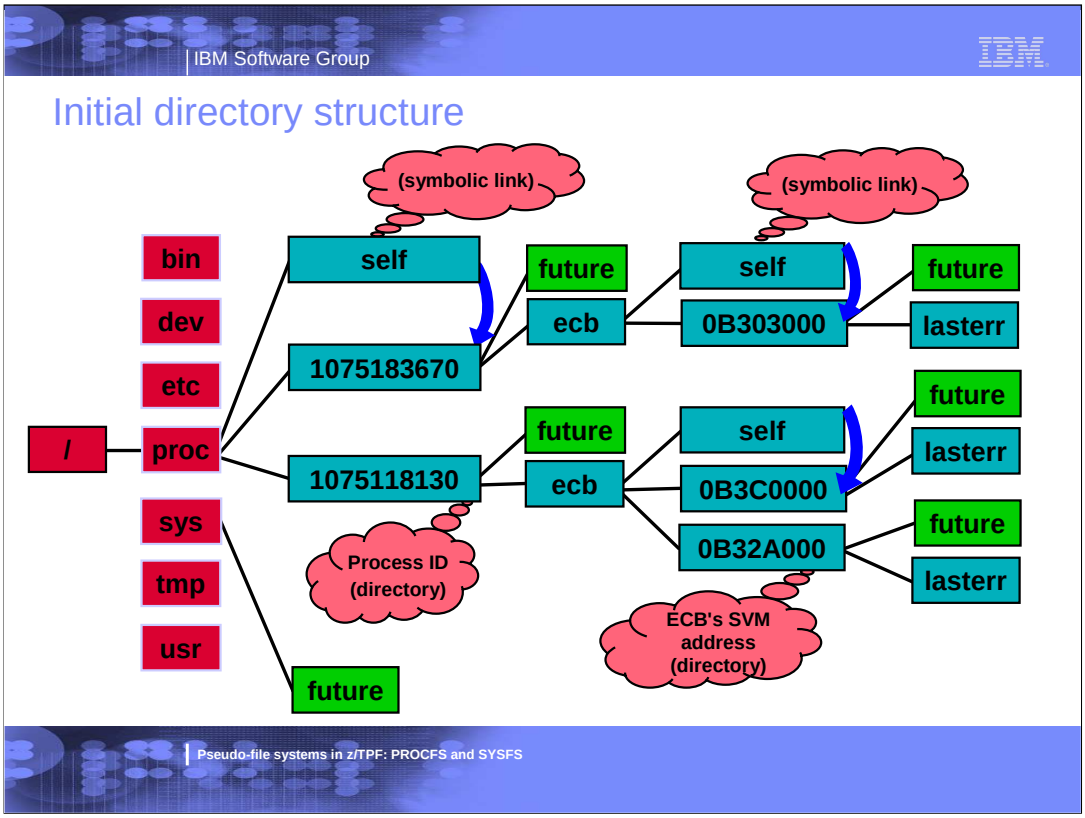
This page describes pseudo-file systems.



Unlike other file systems, the PROCFS can only be mounted on the /proc subdirectory. This is a mount point directory that exists for the sole purpose of mounting the PROCFS file system. The /sys subdirectory serves the same purpose for the SYSFS file system. Only one instance of a PROCFS may exist. Only one instance of a SYSFS may exist.



The file system infrastructure provided as part of initial support is the SYSFS file system code and code for a root directory target which allows the (initially empty) SYSFS to be mounted.



In the above picture:

•The /sys mount point is shown. The SYSFS file system is mounted here, initially empty except for it's root which is mounted on /sys.

•The /proc mount point is shown. The PROCFS file system is mounted here.

•It is shown with the top level subdirectories which represent active processes. The "target" (more on targets later) for the top level <pid> directory is implemented as a program that represents, generically, all process ids. The angle brackets < > around a target name indicate that it is a variable named target with this ability to represent a variety of named files or directories using one generic program (a.k.a. one generic target). Collectively you can refer to the top level directories in the PROCFS as the "<pid>" target, but in practice, multiple directories will exist. One directory for each active process in the system, and each of their names will be a numeric process ID (for example the 1075183670 directory in the picture).

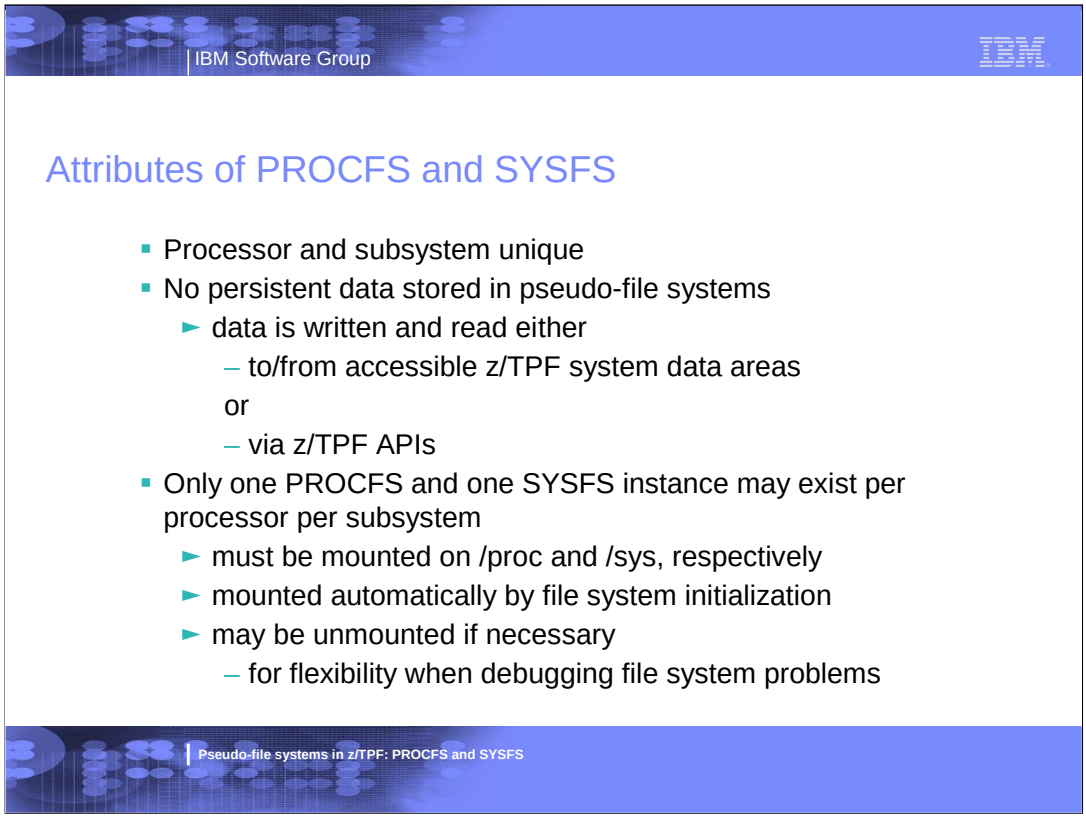
•Also at the top level is the "self" target. It is a fixed name target which is a symbolic link to the current process.

►An "ecb" target exists under each "<pid>". It is a directory implemented as a fixed name "target".

•The "<sva>" target exists under each "ecb" directory. It is implemented as a program that represents, generically, all ecbs for the process identified by the higher level "<pid>" directory. Collectively, you can refer to the this target as the "<sva>" target, but in practice, multiple directories will exist. One directory for each ecb in the process identified by the higher level "<pid>" directory. Each of their names will be a numeric ecb SVM address (for example the 0B32A000 directory in the picture).

►Another "self" target exists under the "ecb" directory. It is a fixed name target which is a symbolic link to the current ecb.

►A "lasterr" file exists under each "<sva>". It is a file implemented



To create or re-create the entire file system from scratch (destroys the TFS and rebuilds it) use ZFINT ON BP. This can also be accomplished by reinitializing Collection Support (which causes the above initialization after rebuilding Collection Support) using ZOODB INIT. This will cause an initial directory structure to be created in the TFS, including the /proc and /sys mount point directories. Also, as part of this initialization, the PROCFS and SYSFS will be mounted on on those mount points. Since most customers will want to avoid this initialization, the creation of the /proc and /sys directories and the initial mounting of the two file systems should be done after applying the PROCFS/SYSFS APAR. After the first mount the mounts are registered in the mtab file and therefore file system remount processing will remount them automatically on each IPL.

Although the PROCFS and SYSFS should not typically be unmounted, the ability is provided to aid in debugging and fixing file system problems.

Example - list all active processes

- The following C program lists all of the active processes in the system without implementing TPF specific code to access the z/TPF data structures directly:

```
DIR* dir = NULL;
struct dirent* dirent = NULL;
dir = opendir( "/proc" );
while( ( dirent = readdir( dir ) ) != NULL )
    printf( "%s", dirent->d_name );
closedir( dir );
```

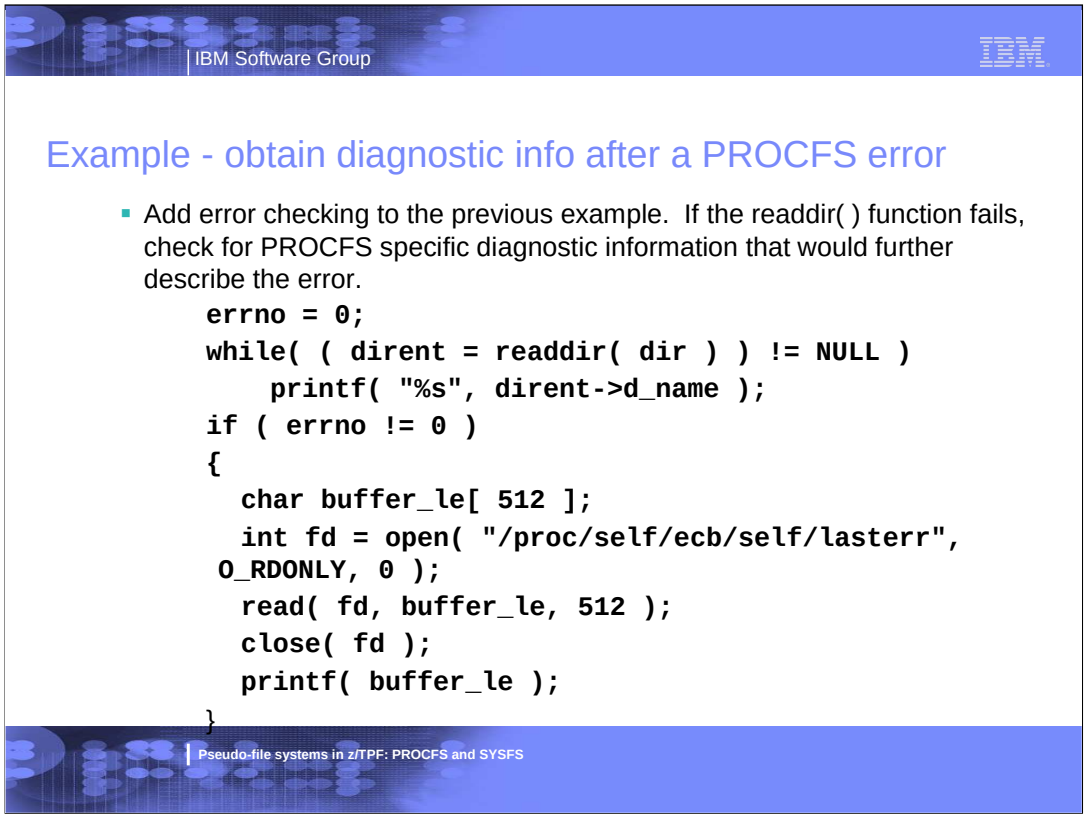
- To obtain similar results from the command line, use:

```
zfile ls /proc
```

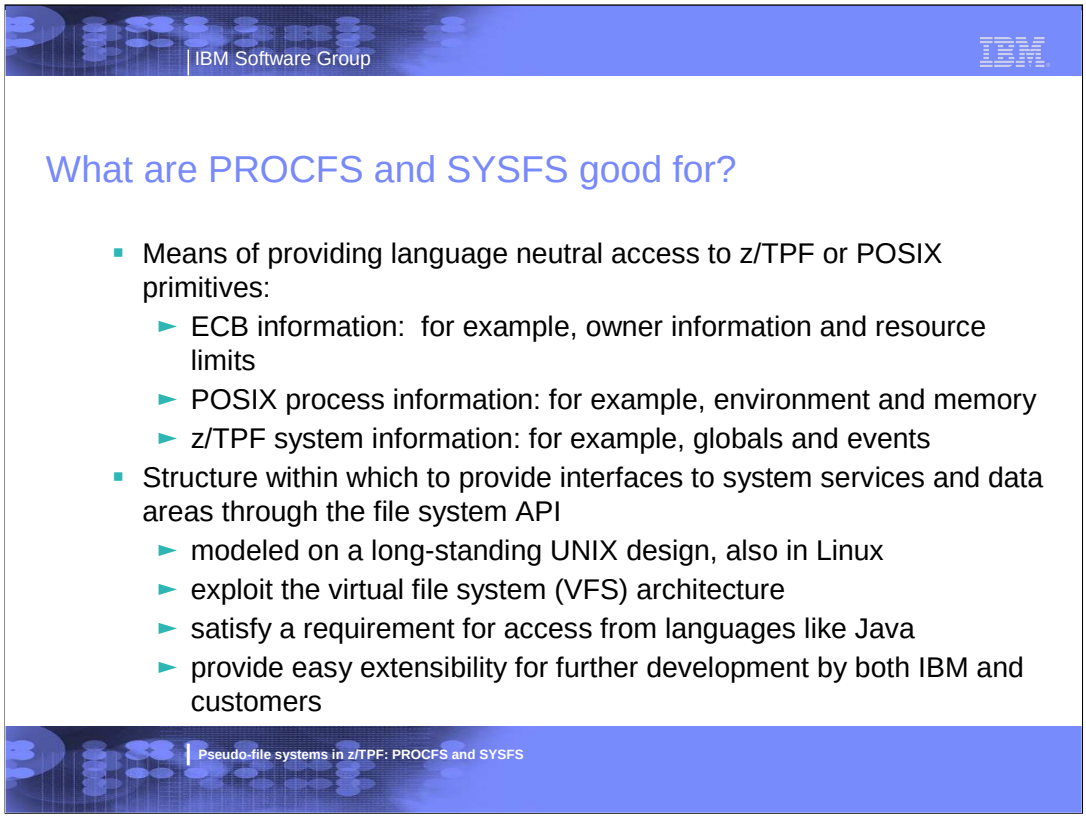
- Or there's the more traditional command line alternative:

```
zfile ps
```

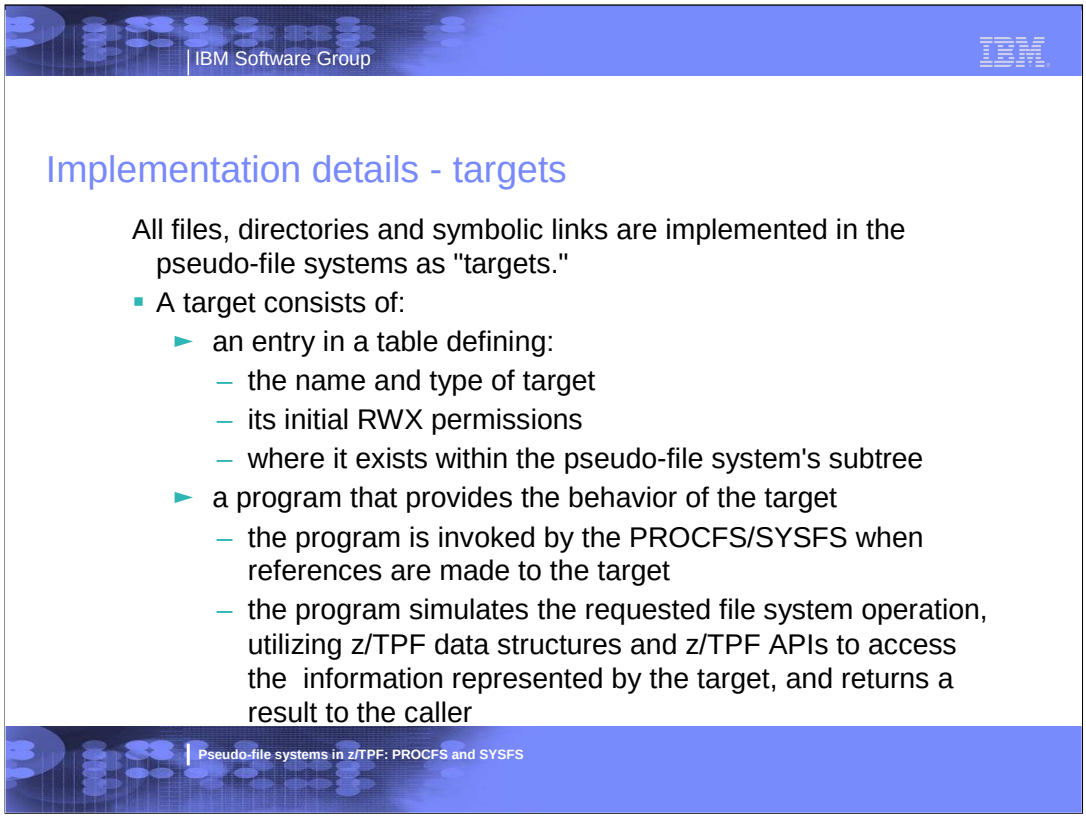

This example lists all of the active processes in the system.



This example obtains diagnostic information after a PROCFS error.



This page describes advantages of PROCFS and SYSFS.



This page describes implementation details - targets.

More about targets

- New targets may be added by both IBM and customers
  - IBM will be announcing additional PROCFS and SYSFS targets at a later time
    - For example eownrc may be implemented under the directory: /proc/<pid>/ecb/<sva>/owner
  - Users can implement a new target by
    - adding a new table entry to the UPRC or USYS user exit and
    - creating a new program that defines the behavior of the target which is referred to by the user exit table entry
- For more information on creating new targets, search for the heading "Creating the target program" in the Information Center: http://www.ibm.com/tpf/pubs/tpfpubs.htm

This page contains more information about targets.

Summary

- PROCFS and SYSFS support added by PJ31637
- Provide access to kernel data through the file system API
- Readily extensible by both IBM and customers
- Documented in detail in the Information Center
  - z/TPF Database User's Guide
  - z/TPF Process Pseudo-File System and System Pseudo-File System Targets

Trademarks

Notes
Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

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.