



SAP BASIS Introductory Training Program

Day 11: Agenda

Introduction to Unix and Vi Editor

Break

Database Overview

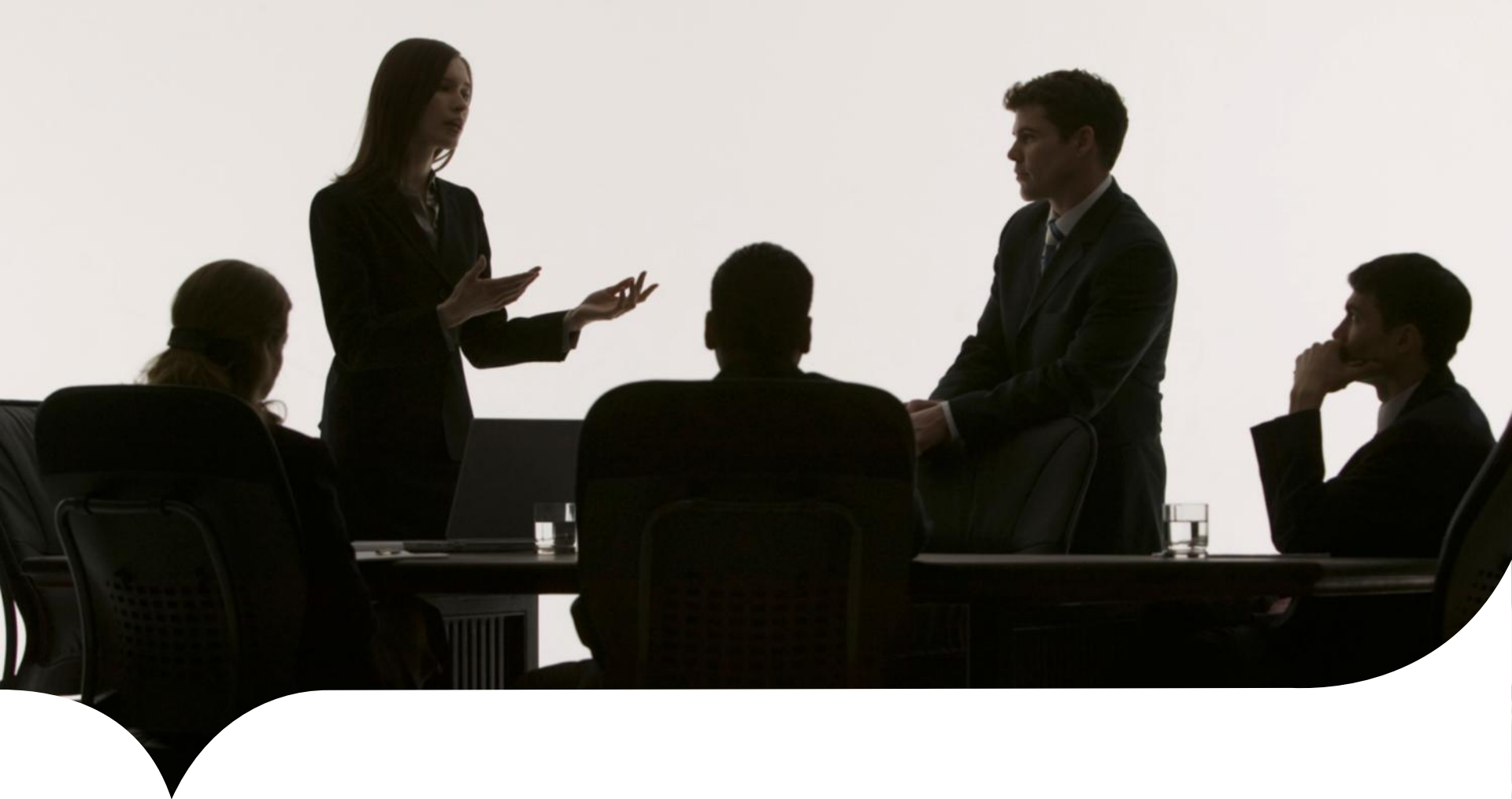
Lunch Break

Database Processes & Structure

Break

Database Management

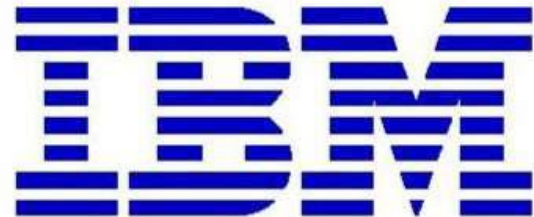
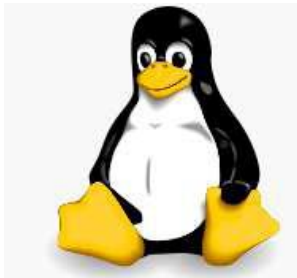
Exercise Break Out Session



Introduction to UNIX

Introduction to UNIX

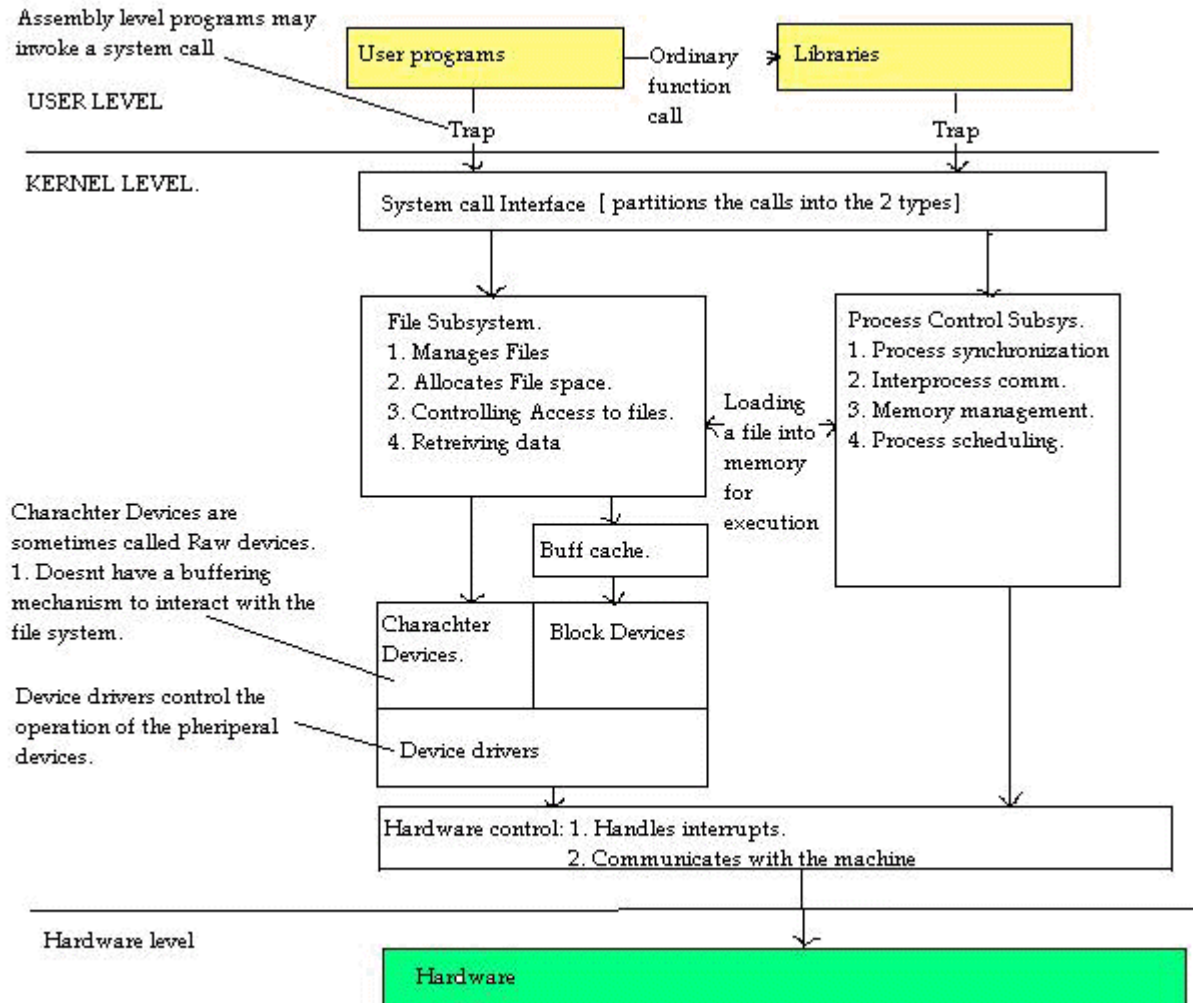
- A fully featured modern operating system. It is available in a variety of flavors. It is comprised of simple tools that perform a single function well. These tools can be used together to perform complex tasks
- UNIX Flavors
 - Solaris: SUN Microsystems
 - HP-UX: HP Version of UNIX
 - AIX: IBM Version of UNIX
 - LINUX: Open Source – Red Hat, SUSE



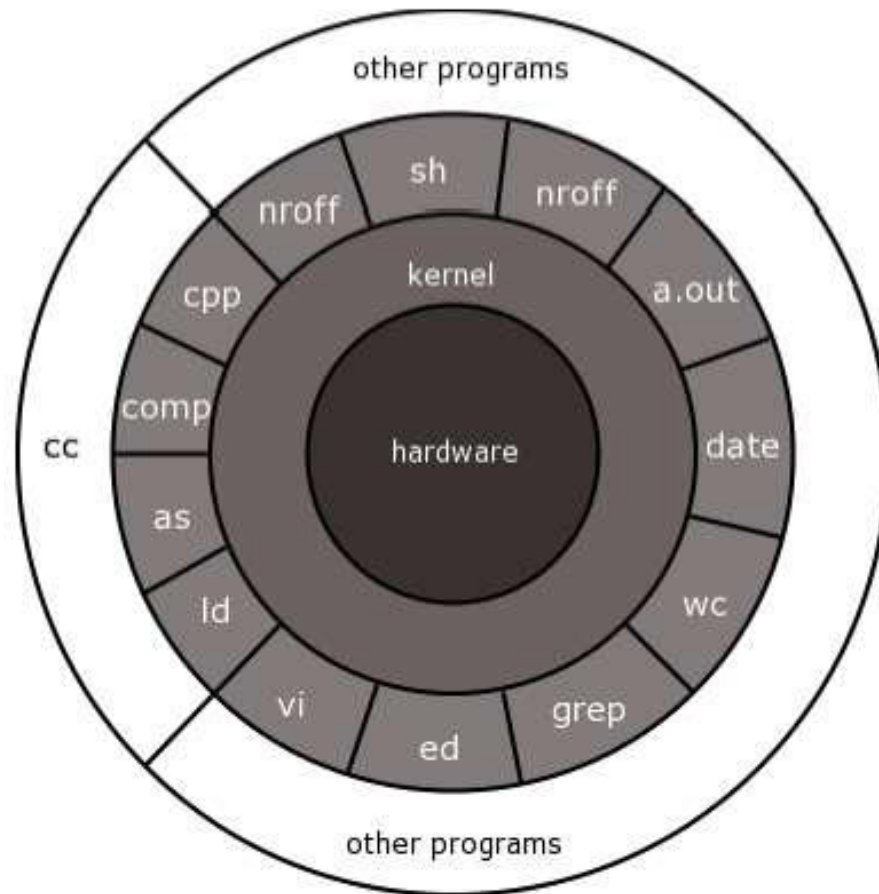
Components of UNIX

- Kernel
 - Manipulate file on disk
 - Start and run programs concurrently
 - Assign memory and other resources
 - Prevent accessing H/W directly
 - Kernel tools → “System Calls”
- System Programs: needed to get the system working: Telnet
- Application Programs: for getting useful things done -> Word Processing
- Compilers and libraries □ GCC
- Device drivers

UNIX Architecture



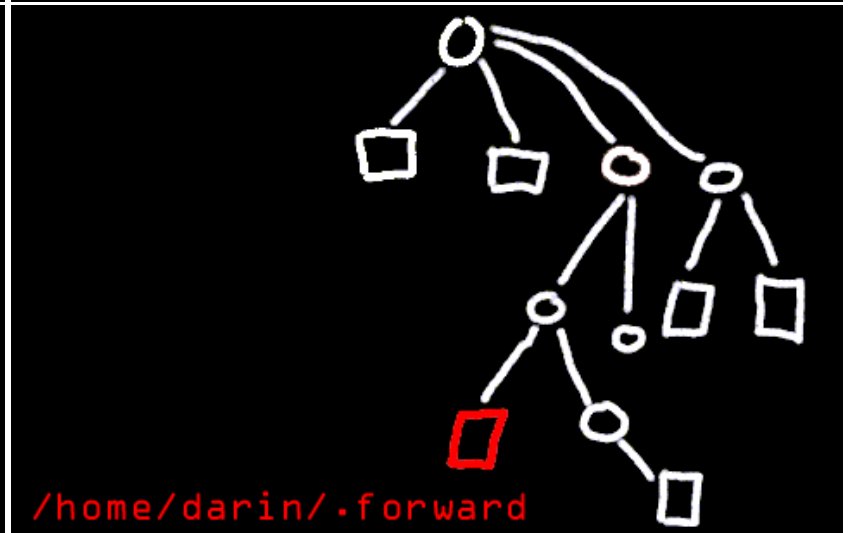
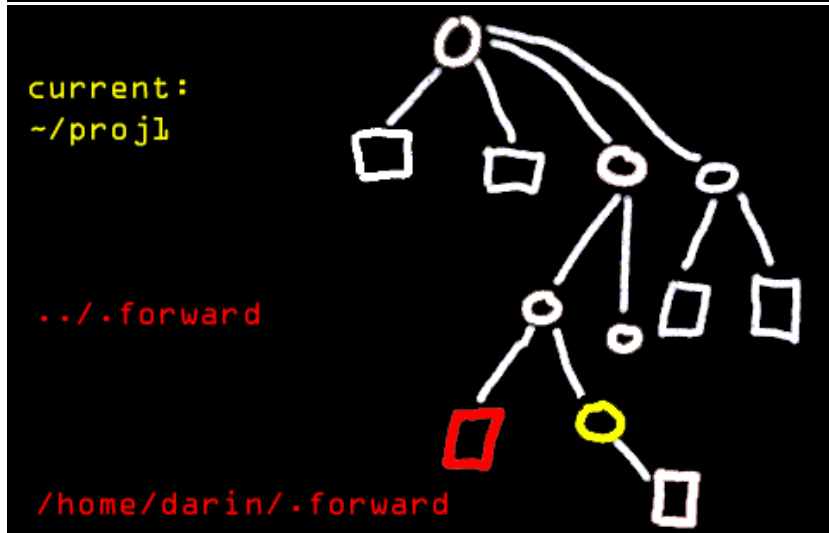
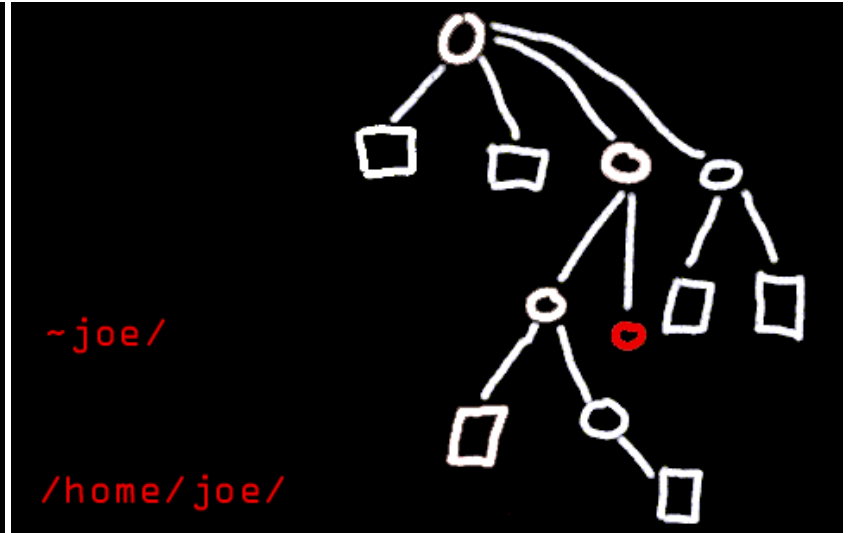
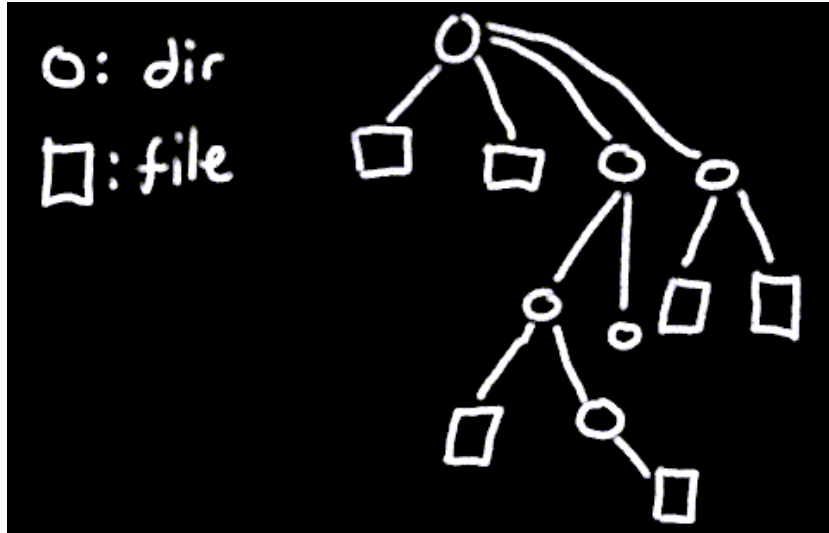
UNIX – Kernel Diagram



UNIX Shell

- The shell is the program that runs when you log in. It prints the prompt and reads what you type, invokes programs, etc.
- It is your window to the Unix world.
- use “chsh <new shell>” to change your shell
- Bourne Shell (C – Shell) or called csh
- Korn Shell (K – Shell) or called ksh

UNIX Directory Structure



UNIX Commands

- Unix commands are normally used in the form:
 - `<command> <one or more options> <arguments>`
 - Where the options are generally included with the `–` sign.
 - Example
- `ls -l -a /usr/local/bin/files`
- `ls -la /usr/local/bin/files`
- If you want to do the command on multiple files you can use the `*` “wild card” character.
 - Examples
- `$ mv *.pl programs/perl/`
- `$ ls data/*`

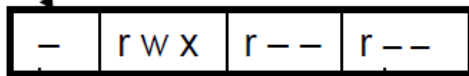
Unix Commands

- `man` (Purpose: Gives you the manual page for a given command)
 - Example
 - `$ man pwd` (This will give description of command `pwd`, i.e. path of working directory)
- `id` (Purpose: Tells you your username (!) and what group you belong to)
- `exit` (or `ctrl-d`) (Purpose: Logs you out from the system)
- `mkdir` (Purpose: Makes a directory)
 - Common options `-p` (creates all the sub-directories in a path if they don't exist)
- `cd` (Purpose: Changes directory. With no arguments changes to home directory)
- `mv` (Purpose: Moves or renames a file or directory)
 - Common options
 - `-i` (Asks confirmation before overwriting another file or directory)
- `rm` (Purpose: Deletes or renames a file)
 - Common options
 - `-i` (Asks confirmation first)
 - `-r` (deletes all sub-directories of a directory (VERY DANGEROUS))
- `cp` (Purpose: Makes a copy of a file or directory)
 - Common options
 - `-i` (Asks confirmation first if overwriting another file)
 - `-r` (copies all files of all sub-directories of a directory)
- `ls` (Purpose: Lists files and directories)
 - Common options
 - `-t` (sort by modification time)
 - `-l` (long format, gives all details of the file (very useful))
 - `-a` (shows file beginning with `.` (not visible with just `ls`))

File Permission

```
$ ls -l sapdba
```

```
-rwxr--r-- 1 oradba  orapp1  30720 Mar 13 10:10 sapdba
```



Other users permission. Read, Write, Execute

Group permission. Read

File owners permission. Read

Special permission e.g. sticky bit, symbolic links

- Command :chmod
- Purpose: Changes the permissions of a file or directory. Only the owner of a file, or root, can change the permissions.
- Common options: -R, changes all the permissions in a directory, including sub-directories

```
$ chmod u+x myprog.pl (make file executable for owner)
```

```
$ chmod g+w,o-w seq.dat (write access for group, no write for others)
```

```
$ chmod +r *.fasta (add read access for all)
```

```
$ chmod 777 *.prog (octal notation, here = +rwx for all; r=4,w=2,x=1)
```



Using the Vi EDITOR

Editing with vi

- vi is the standard Unix text editor and is present on every Unix system
- vi has 3 modes:
 - Command mode
- For manipulating and moving through the the text
 - Line mode
- For special commands and interacting with Unix.
 - Insert mode
- For entering text, i.e. writing programs, entering data, etc.

vi – Command Mode

- Command mode – the usual and initial mode (i.e. when starting vi)
- Commands include
 - ←↑↓→ arrow keys move the cursor
 - hjkl same as arrow keys
 - x delete a character
 - dw delete a word
 - dd delete a line
 - 3dd delete 3 lines
 - u undo previous change
 - ZZ exit vi, saving changes

vi – Line Mode

- Line mode – entered by typing :, / , ? or ! .
- Commands include
 - :q! save file, discarding changes
 - :q quit
 - :e filename edit a new file
 - :w filename write with new filename
 - :wq write file and quit
 - :!cmd run Unix command
 - /string look for string
- RETURN executes command and returns to command mode

vi – Insert Mode

- Insert mode – entered by typing any of the following in command mode
 - a append after cursor
 - i insert before cursor
 - o open line below
 - O open line above
 - Rtext replace with text
- To exit insert mode, and return to command mode, type <ESC>.

Unix Commands used in SAP

Command	Description
chgrp grp filelist	Change the group that the file(s) belong to.
chmod 777 filelist	Change the rights of the file(s). 777: owner-world-user. 7: read-write-delete (=111)
chown user filelist	Change the owner of the file(s)
cp file1 file2	Copy file1 to file2
cp /dev/null file	Clear all contents of file
errpt	Hardware error-report (IBM-unix)
ls -ltr <dir>	Show filelist, -parameters: l: show fileattributes , t: sort by time, r: reverse sortorder
more <dir>	
mv file1 file2	Move command
ps -ef	Show all running processes, -e, -f
vi	The famous vi-editor
kill <PID>	Kill a process as if it is cancelled. Child processes are killed as well.
kill -1 <PID>	Kill a process as if the user is logging out. Child processes are killed as well.
kill -9 <PID>	Kill a process as brutal as possible. Child-processes are not touched.
kill -USR2 <PID>	Increases TRACE by 1
kill -USR1 <PID>	Decreases TRACE by 1. When TRACE = 0 all open trace files are closed and can be cleared using „cp /dev/null filename“.
lslv	Show the logical volumes
lsvg	Show the volume groups
sapfpgr check pf=<profile>	Checks the profileparameter. Do they exist and do they have correct values.
set VAR VAL	set environment variable VAR to value VAL
printenv	prints the environmentvariables
ps -ef grep xxx	Monitor processes with name containing xxx
ps -u <username>	Monitor processes run by user <username>

Summary

- Client Administration
 - Client Copy (Local, Remote, Transport)
 - Client Copy Tools
- Introduction to Unix
 - Architecture
 - C & K Shell
 - Commands
 - Editor vi
 - Commands used in SAP

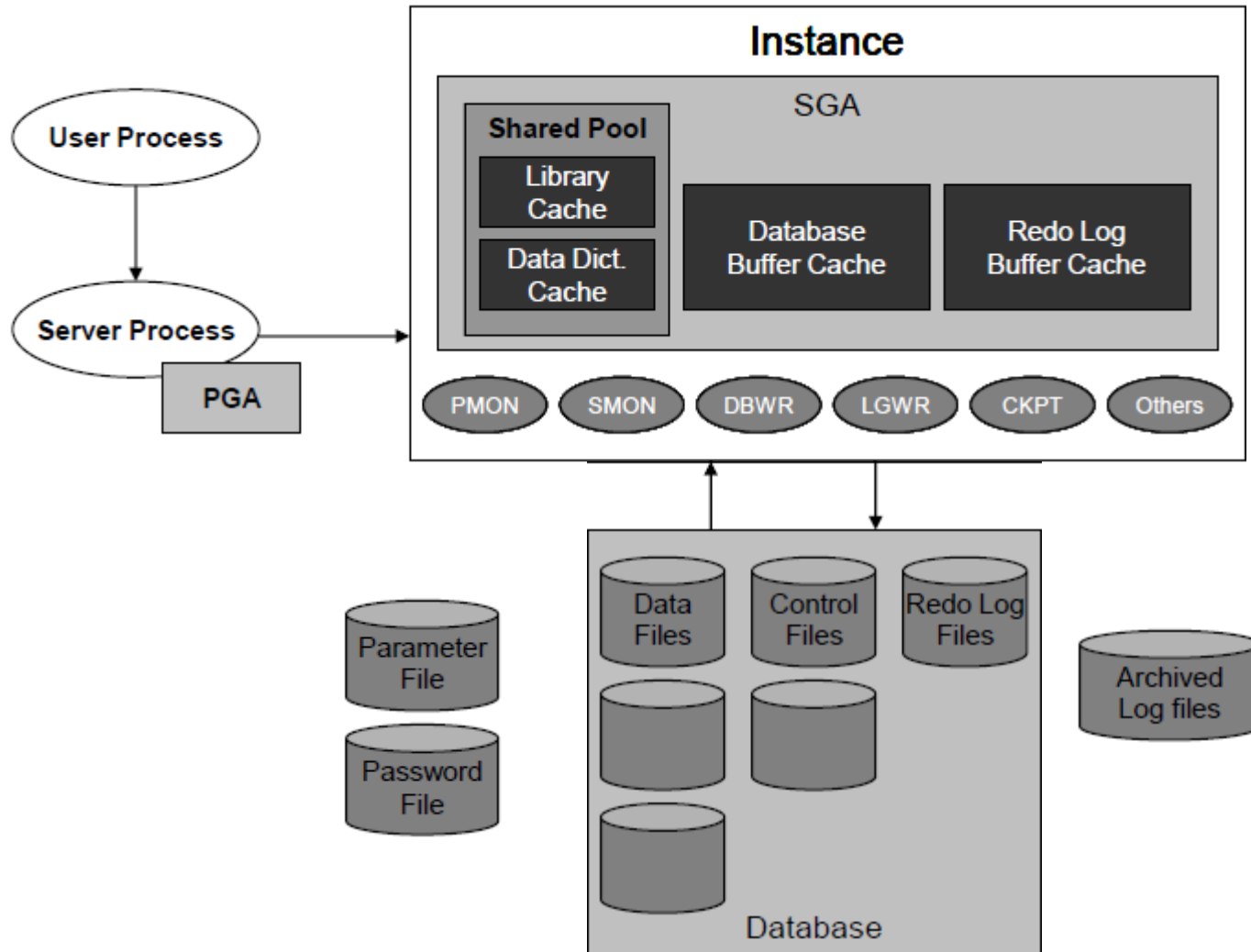


Break



Database Overview

Oracle Architecture



Memory Structure

- System Global Area (SGA):
 - Allocated at instance startup and is fundamental component of an Oracle instance
 - SGA consists of several memory structures
 - Shared Pool
 - Database buffer cache
 - Redo log buffer
 - Other structures (e.g. lock & latch management, statistical data)
 - Two optional memory structures that can be configured with in the SGA
 - Large Pool
 - Java Pool
 - SGA is dynamic and sized using SGA_MAX_SIZE
- Program Global Area (PGA):
 - Allocated when the server process is started
 - PGA is memory reserved for each user process that connects to an Oracle instance

Shared Pool

- Shared Pool

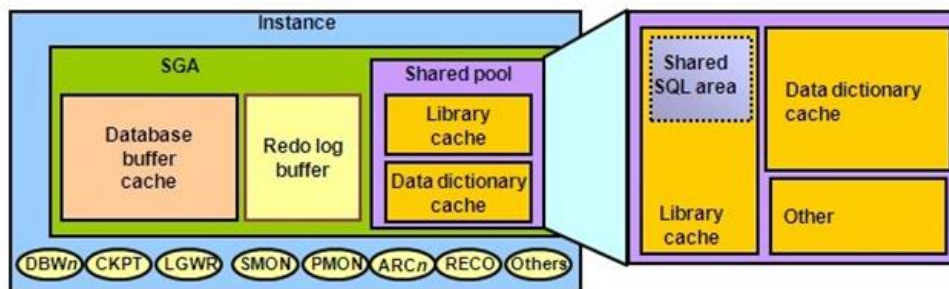
- Shared Pool is used to store most recently executed SQL statements and the most recently used data definitions
- It consists of two key performance-related memory structures
 - Library cache
 - Data Dictionary cache
- Sized by the parameter SHARED_POOL_SIZE

- Library Cache

- Library Cache stores information about the most recently used SQL & PL/SQL statements
- Uses **Least Recently Used** algorithm (LRU)
- Consists of two structures: Share SQL area and Shared PL/SQL area

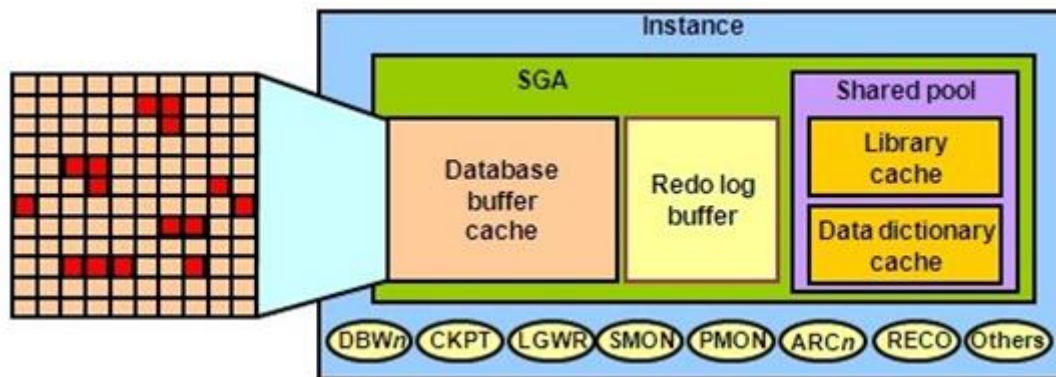
- Data Dictionary Cache

- Data Dictionary Cache is collection of most recently used definitions in database
- Information about database files, tables, indexes, columns, users, privileges, and other database objects
- Caching data dictionary information into memory improves response time on queries



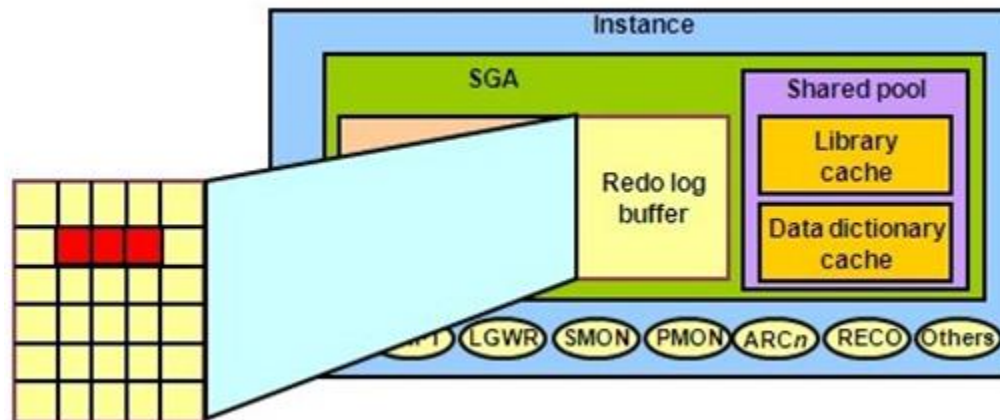
Database Buffer Cache

- The database buffer cache stores copies of data blocks that have been retrieved from the data files.
- It enables great performance gains when obtaining and updating data.
- It is managed through a least recently used (LRU) algorithm.
- DB_BLOCK_SIZE determines the primary block size.
- Consists of independent sub-caches
 - DB_CACHE_SIZE
 - DB_KEEP_CACHE_SIZE
 - DB_RECYCLE_CACHE_SIZE
- Database buffer cache can be dynamically resized to grow or shrink using ALTER SYSTEM.
- DB_CACHE_ADVICE can be set to gather statistics for predicting different cache size behavior.



Redo Log Buffer Cache

- The redo log buffer cache records all changes made to the database data blocks.
- Its primary purpose is recovery.
- Changes recorded within are called redo entries.
- Redo entries contain information to reconstruct or redo changes.
- Size is defined by LOG_BUFFER.



Large and Java Pool

- Large Pool

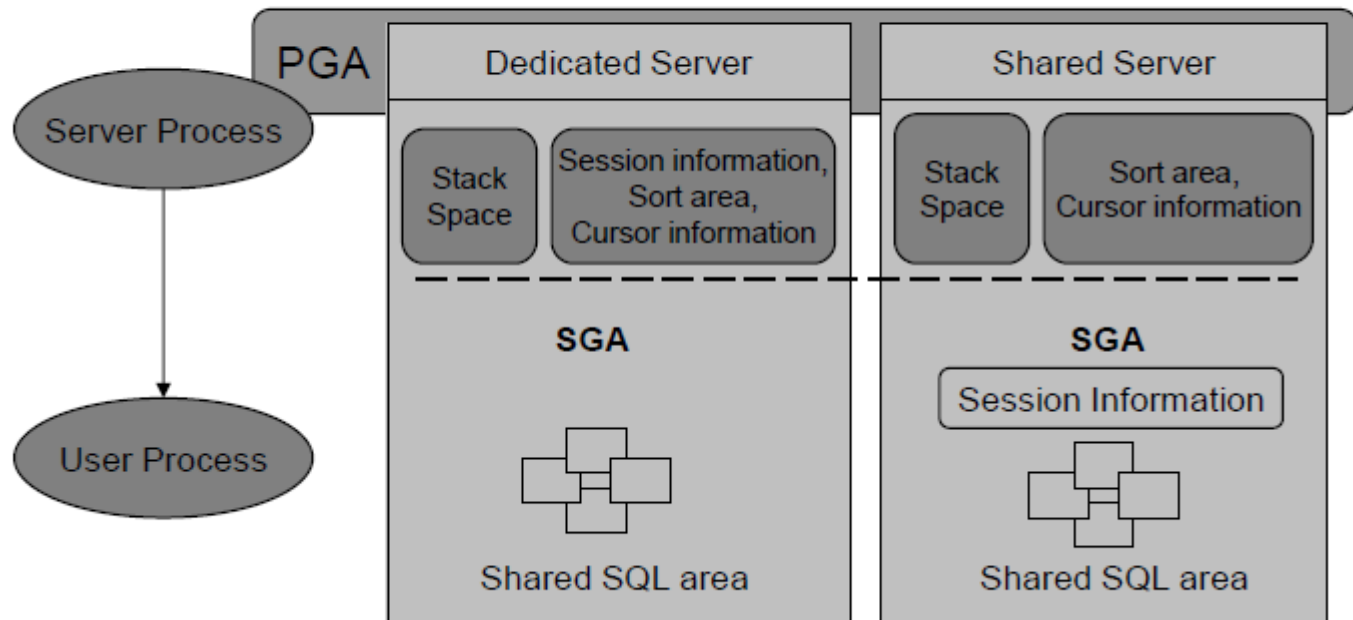
- The large pool is an optional area of memory in the SGA configured only in a shared server environment.
- It relieves the burden placed on the shared pool.
- This configured memory area is used for session memory (UGA), I/O slaves, and backup and restore operations.
- Unlike the shared pool, the large pool does not use an LRU list.
- Sized by `LARGE_POOL_SIZE`.

- Java Pool

- The Java pool services the parsing requirements for Java commands.
- Required if installing and using Java.
- It is stored much the same way as PL/SQL in database tables.
- It is sized by the `JAVA_POOL_SIZE` parameter.

Program Global Area

- The PGA is memory reserved for each user process that connects to an Oracle database





Lunch Break



Database Processes and Structure

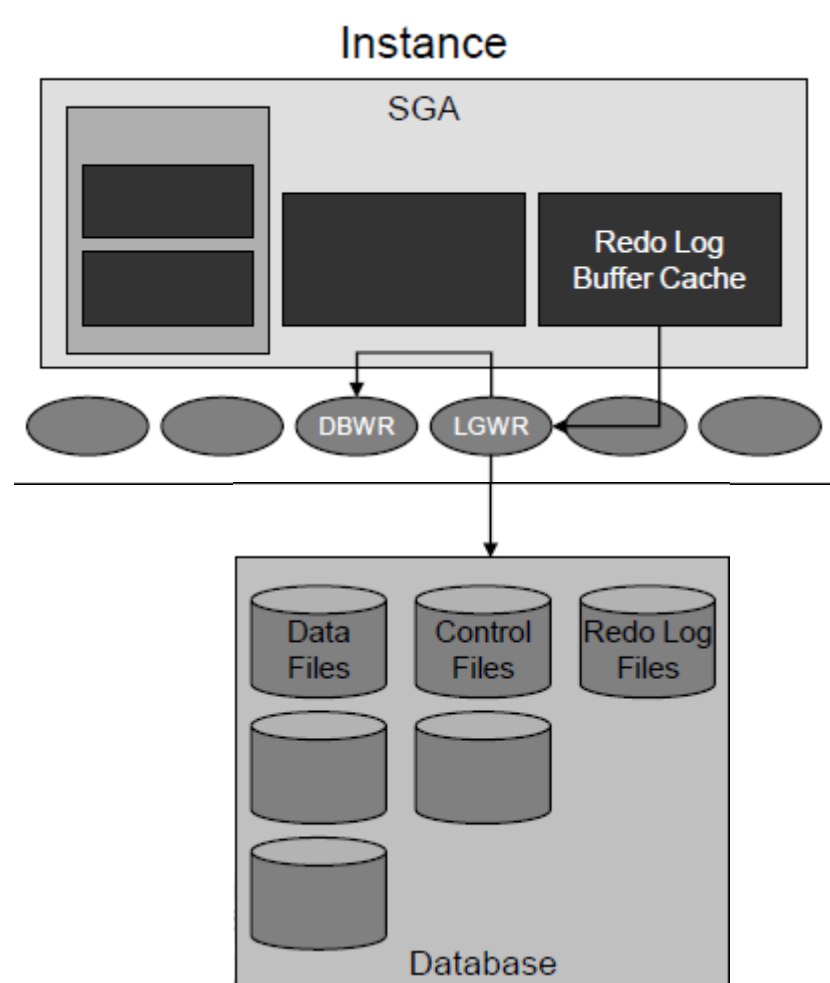
Oracle Background Process

- The relationship between the physical and memory structures is maintained and enforced by Oracle's background processes
- Mandatory Processes
 - Database Writer (DBWn)
 - Log Writer (LGWR)
 - System Monitor (SMON)
 - Process Monitor (PMON)
 - Checkpoint (CKPT)
 - Archive (ARCn)
- Optional Processes
 - Recovery (RECO)
 - Instance Locks
 - Global Locks
 - Remote Locks
 - Dispatcher
 - Shared server
 - Parallel Query Slaves
 - Advance Queuing

Log Writer (LGWR)

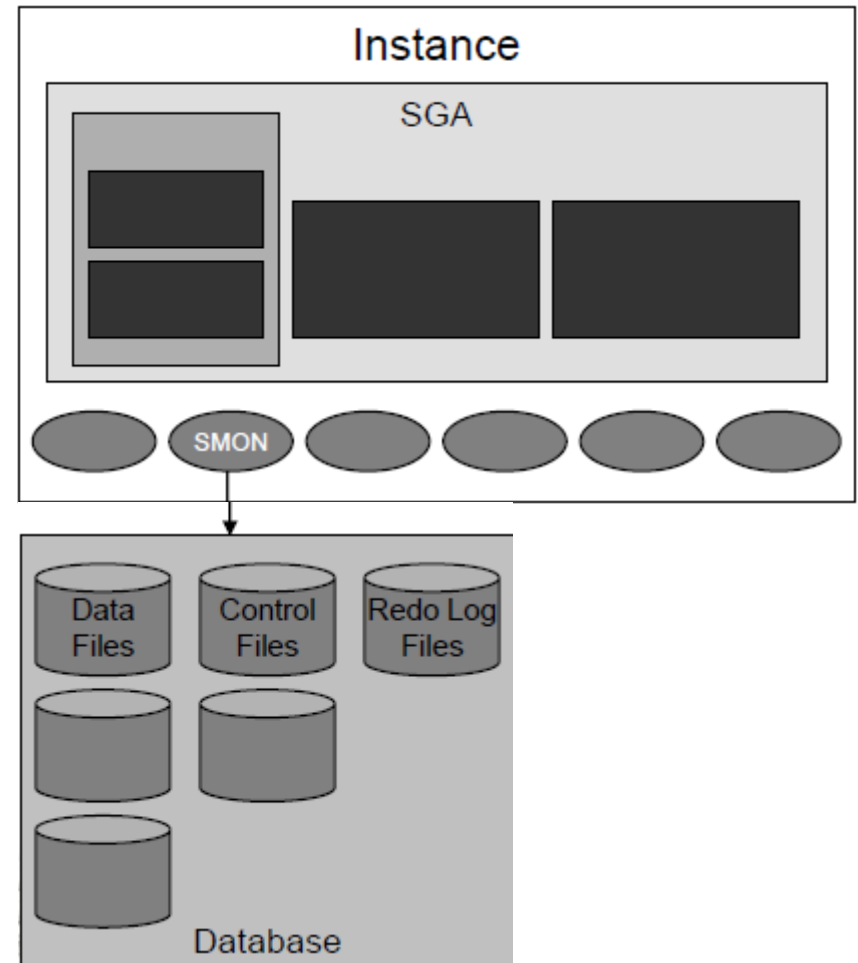
- LGWR writes:

- At commit
- When one-third full
- When there is 1 MB of redo
- Every 3 seconds
- Before DBWn writes



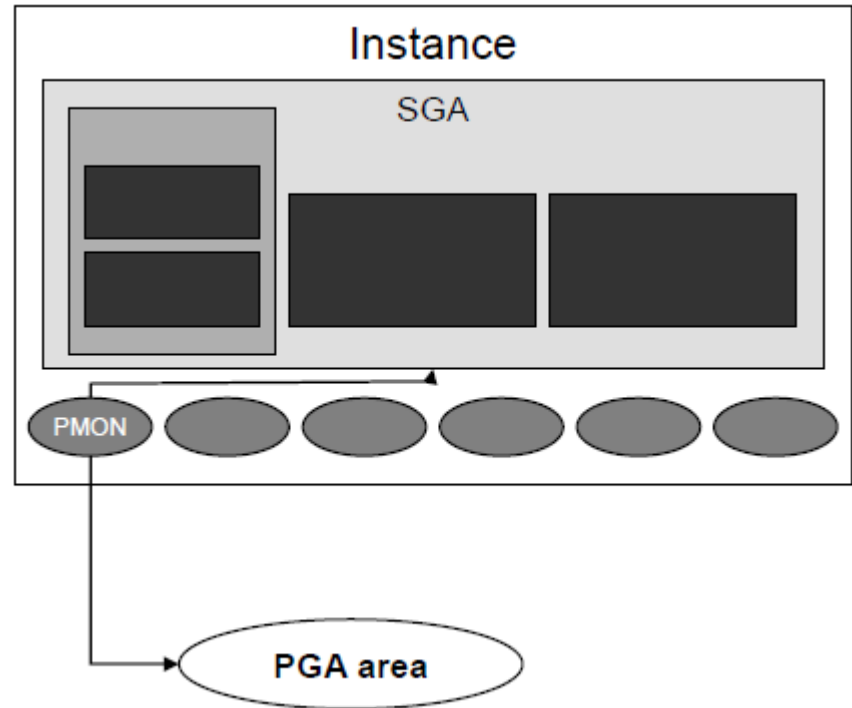
System Monitor (SMONM)

- Instance recovery:
 - Rolls forward changes in the redo logs
 - Opens the database for user access
 - Rolls back uncommitted transactions
- Coalesces free space ever 3 sec
- De-allocates temporary segments



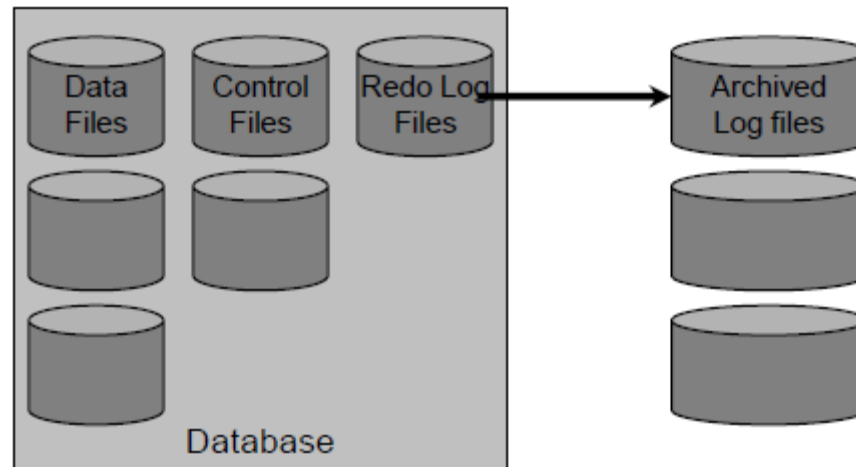
Process Monitor (PMO)

- Cleans up after failed processes by
 - Rolling back the transaction
 - Releasing locks
 - Releasing other resources
 - Restarts dead dispatchers



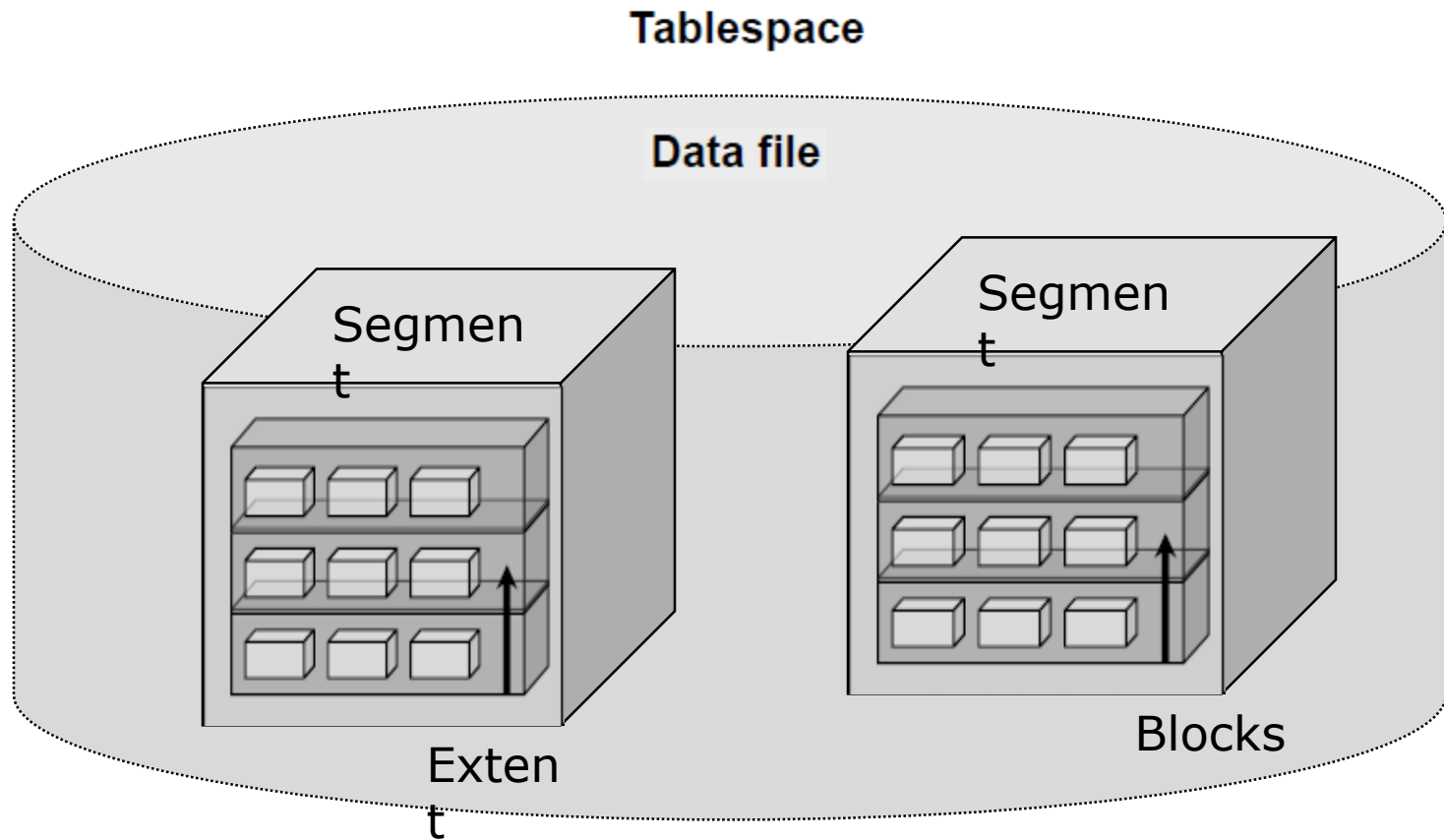
Archive (ARCn)

- Optional background process
 - Automatically archives online redo logs when ARCHIVELOG mode is set
 - Preserves the record of all changes made to the database

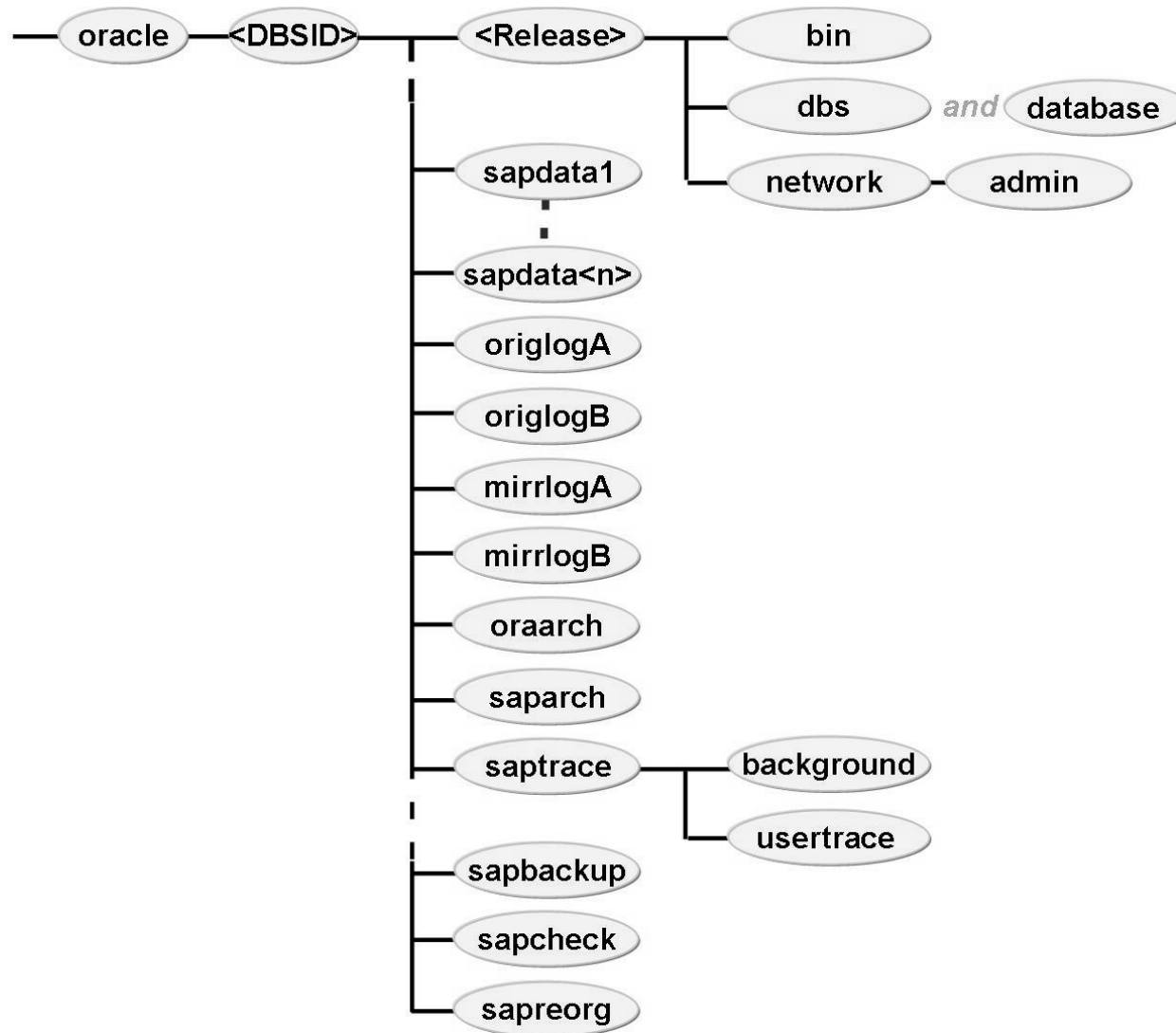


Logical Structure

- The logical structure of the Oracle architecture dictates how the physical space of a database is to be used.
- A hierarchy exists in this structure that consists of tablespaces, segments, extents, and blocks.

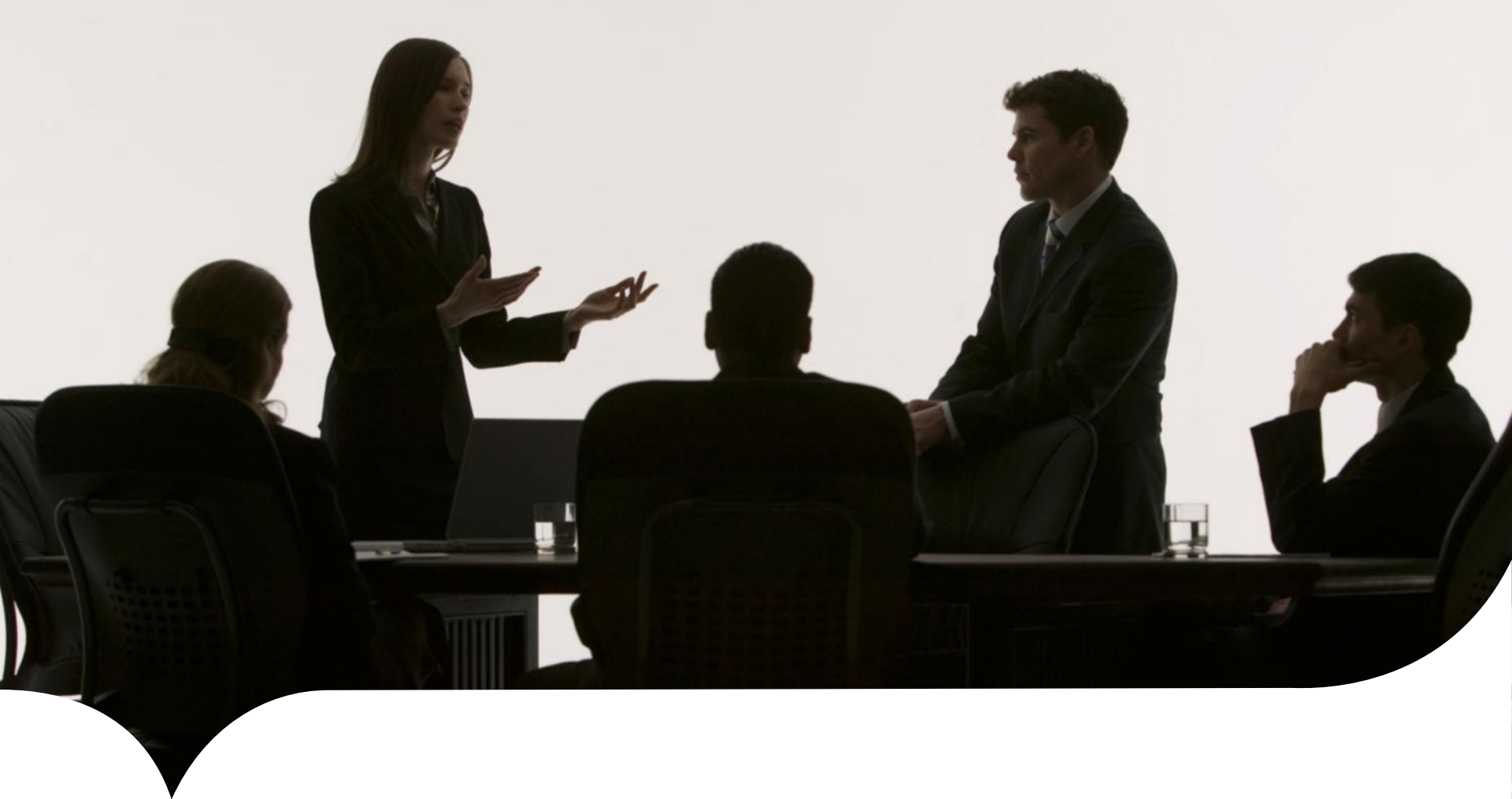


Oracle Directory Structure in SAP



Temporary Tablespace

- Used for sort operations
- Cannot contain any permanent objects
- Locally managed extents recommended
- It cannot be dropped until after a new default is made available.
- It cannot be taken offline.
- You cannot alter the default temporary tablespace to a permanent tablespace.
- Default Temporary Tablespace
 - Allows you to specify a databasewide default temporary tablespace
 - Eliminates the use of the SYSTEM tablespace for storing temporary data
 - Can be created using the CREATE DATABASE or ALTER DATABASE command.
 - When created with the CREATE DATABASE command, the default temporary tablespace is locally managed



Exercise & Break Out Session

Exercise

- Logon to the operating system level with the userid/password provided by the instructor
- Instructor Steps Steps to demonstrate start/stop using BRTOOLS
 - Stop the SAP application instance using the <SID>adm user
 - Switch to the ora<SID> user
 - Start BRTOOLS from the command prompt
 - Execute the stop command
 - Check if any oracle processes exist using the `ps -ef | grep ora` command
 - Start the Oracle Instance
- Trainee steps for 2 Groups to be performed one at a time :
- Group 1
 - Check the tablespaces and datafiles for the main PSAP tablespace
 - Add a 2M datafile to this tablespace using the appropriate tablespace command
- Group 2
 - Resize the datafile created by Group 1 by 2 M
 - Note the change in filesystem space availability after the activities