

Oracle Background Processes

Database Writer

Process Name: **DBW0** through DBW9 and DBWa through DBWj

Max Processes: 20

This process writes the dirty buffers present in the database buffer cache to datafiles. The process does multi-block writing to disk asynchronously. One database writer process is adequate for most systems; more can be configured if necessary. The initialization parameter, DB_WRITER_PROCESSES, specifies the number of database writer processes to start.

The DBWn process writes dirty buffer to disk under the following conditions:

1. When a checkpoint is issued. Please see checkpoint process below.
2. When a server process cannot find a clean reusable buffer after scanning a threshold number of buffers.
3. Every 3 seconds. See Followup on DBWn DBWn 3 second write

Database writer must always be running for an instance.

Log Writer

Process Name: **LGWR**

Max Processes: 1

The log writer process writes data from the redo log buffers to the redo log files on disk.

The writer is activated under the following conditions:

1. When a transaction is committed, a System Change Number (SCN) is generated and tagged to it. Log writer puts a commit record in the redo log buffer and writes it to disk immediately along with the transaction's redo entries. Changes to actual data blocks are deferred until a convenient time (Fast-Commit Mechanism).
2. Every 3 seconds.
3. When the redo log buffer is 1/3 full.
4. When DBWn signals the writing of redo records to disk. All redo records associated with changes in the block buffers must be written to disk first (The write-ahead protocol). While writing dirty buffers, if the DBWn process finds that some redo information has not been written, it signals the LGWR to write the information and waits until the control is returned.

Log writer will write synchronously to the redo log groups in a circular fashion. If any damage is identified with a redo log file, the log writer will log an error in the LGWR trace file and the system Alert Log. Sometimes, when additional redo log buffer space is required, the LGWR will even write uncommitted redo log entries to release the held buffers. LGWR can also use group commits (multiple committed transaction's redo entries taken together) to write to redo logs when a database is undergoing heavy write operations.

The log writer must always be running for an instance.

Archiver

Process Name: ARC0 through ARC9

Max Processes: 10

The ARCN process is responsible for writing the online redo log files to the mentioned archive log destination after a log switch has occurred. ARCN is present only if the database is running in archivelog mode and automatic archiving is enabled. The log writer process is responsible for starting multiple ARCN processes when the workload increases. Unless ARCN completes the copying of a redo log file, it is not released to log writer for overwriting.

The number of Archiver processes that can be invoked initially is specified by the initialization parameter LOG_ARCHIVE_MAX_PROCESSES. The actual number of Archiver processes in use may vary based on the workload.

System Monitor

Process Name: SMON

Max Processes: 1

This process is responsible for instance recovery, if necessary, at instance startup. SMON also cleans up temporary segments that are no longer in use. It also coalesces contiguous free extents in dictionary managed tablespaces that have PCTINCREASE set to a non-zero value.

SMON wakes up about every 5 minutes to perform housekeeping activities. SMON must always be running for an instance.

Process Monitor

Process Name: PMON

Max Processes: 1

This process is responsible for performing recovery if a user process fails. It will rollback uncommitted transactions. PMON is also responsible for cleaning up the database buffer cache and freeing resources that were allocated to a process. PMON also registers information about the instance and dispatcher processes with network listener.

PMON wakes up every 3 seconds to perform housekeeping activities. PMON must always be running for an instance.

Checkpoint Process

Process Name: CKPT

Max processes: 1

Checkpoint process signals the synchronization of all database files with the checkpoint information. It ensures data consistency and faster database recovery in case of a crash.

CKPT ensures that all database changes present in the buffer cache at that point are written to the data files, the actual writing is done by the Database Writer process. The datafile headers and the control files are updated with the latest SCN (when the checkpoint occurred), this is done by the log writer process.

The CKPT process is invoked under the following conditions:

1. When a log switch is done.
2. When the time specified by the initialization parameter LOG_CHECKPOINT_TIMEOUT exists between the incremental checkpoint and the tail of the log; this is in seconds.
3. When the number of blocks specified by the initialization parameter LOG_CHECKPOINT_INTERVAL exists between the incremental checkpoint and the tail of the log; these are OS blocks.
4. The number of buffers specified by the initialization parameter FAST_START_IO_TARGET required to perform roll-forward is reached.
5. Oracle 9i onwards, the time specified by the initialization parameter FAST_START_MTTR_TARGET is reached; this is in seconds and specifies the time required for a crash recovery. The parameter FAST_START_MTTR_TARGET replaces LOG_CHECKPOINT_INTERVAL and FAST_START_IO_TARGET, but these parameters can still be used.
6. When the ALTER SYSTEM SWITCH LOGFILE command is issued.
7. When the ALTER SYSTEM CHECKPOINT command is issued.

Incremental Checkpoints initiate the writing of recovery information to datafile headers and controlfiles. Database writer is not signaled to perform buffer cache flushing activity here.

Lock Monitor

Process Name: LMON

processes: 1

Meant for Parallel server setups, Lock Monitor manages global locks and resources. It handles the redistribution of instance locks whenever instances are started or shutdown. Lock Monitor also recovers instance lock information prior to the instance recovery process. Lock Monitor co-ordinates with the Process Monitor to recover dead processes that hold instance locks.

Lock Manager Daemon

Process Name: LMDn

Max Processes: -

Meant for Parallel server setups, LMDn processes manage instance locks that are used to share resources between instances. LMDn processes also handle deadlock detection and remote lock requests.

Lock processes

Process Name: LCK0 through LCK9

Max Processes: 10

Meant for Parallel server setups, the instance locks that are used to share resources between instances are held by the lock processes.

Block Server Process

Process Name: BSP0 through BSP9

Max processes: 10

Meant for Parallel server setups, Block server Processes have to do with providing a consistent read image of a buffer that is requested by a process of another instance, in certain circumstances.

Queue Monitor

Process Name: QMN0 through QMN9

Max Processes: 10

This is the advanced Queuing Time manager process. QMNn monitors the message queues. Failure of QMNn process will not cause the instance to fail.

Event Monitor

Process Name: EMN0/EMON

Max Processes: 1

This process is also related to Advanced Queuing, and is meant for allowing a publish/subscribe style of messaging between applications.

Recoverer

Process Name: RECO

Max processes: 1

Intended for distributed recovery. All in-doubt transactions are recovered by this process in the distributed database setup. RECO will connect to the remote database to resolve pending transactions.

Job Queue Processes

Process Name: J000 through J999 (Originally called SNPn processes)

Max Processes: 1000

Job queue processes carry out batch processing. All scheduled jobs are executed by these processes. The initialization parameter JOB_QUEUE_PROCESSES specifies the maximum job processes that can be run concurrently. If a job fails with some Oracle error, it is recorded in the alert file and a process trace file is generated. Failure of the Job queue process will not cause the instance to fail.

Dispatcher

Process Name: Dnnn

Max Processes: -

Intended for Shared server setups (MTS). Dispatcher processes listen to and receive requests from connected sessions and places them in the request queue for further processing. Dispatcher processes also pickup outgoing responses from the result queue and transmit them back to the clients. Dnnn are mediators between the client processes and the shared server processes. The maximum number of Dispatcher process can be specified using the initialization parameter MAX_DISPATCHERS.

Shared Server Processes

Process Name: Snnn

Max Processes: -

Intended for Shared server setups (MTS). These processes pickup requests from the call request queue, process them and then return the results to a result queue. The number of shared server processes to be created at instance startup can be specified using the initialization parameter SHARED_SERVERS.

Parallel Execution Slaves

Process Name: Pnnn

Max Processes: -

These processes are used for parallel processing. It can be used for parallel execution of SQL statements or recovery. The Maximum number of parallel processes that can be invoked is specified by the initialization parameter PARALLEL_MAX_SERVERS.

Trace Writer

Process Name: TRWR

Max Processes: 1

Trace writer writes trace files from an Oracle internal tracing facility.

Input/Output Slaves

Process Name: Innn

Max Processes: -

These processes are used to simulate asynchronous I/O on platforms that do not support it. The initialization parameter DBWR_IO_SLAVES is set for this purpose.

Wakeup Monitor Process

Process Name: WMON

Max Processes: -

This process was available in older versions of Oracle to alarm other processes that are suspended while waiting for an event to occur. This process is obsolete and has been removed.

Reference:

http://www.databasejournal.com/features/oracle/article.php/10893_3350331_3/Oracle-Background-Processes.htm