

# LUND

PERFORMANCE SOLUTIONS

WHAT YOU NEED TO KNOW. WHEN YOU NEED TO KNOW IT.

LUND  
PERFORMANCE SOLUTIONS

LUND  
PERFORMANCE SOLUTIONS

LUND  
PERFORMANCE SOLUTIONS

**LUND**  
PERFORMANCE SOLUTIONS

Slide 2

WHAT YOU NEED TO KNOW. WHEN YOU NEED TO KNOW IT.

## Dispatcher Queues Management and Performance Implications

**D. Scott Pierson**

# Queueing



- Definitions
- Queue Rules
- Dispatcher
- Preemption
- CPU
- Tune Command
- Process

## Queueing – Definitions

- **Process:** a unique execution of a program by a user
- **Prompt Response:** the time it takes from the point a user hits the return/enter key until the user can once again enter data onto the terminal
- **First Response:** the time it takes from the point a user hits the return/enter key until the time the user sees the first character back on the console

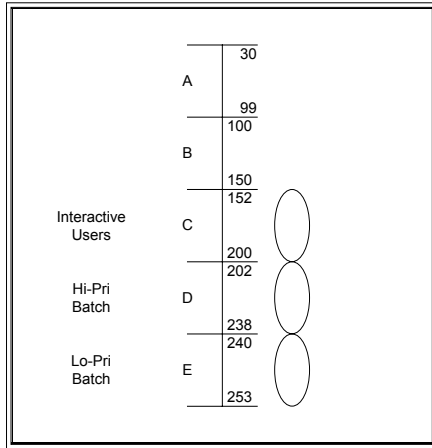
## Queueing – Queue Rules

- The AS Queue is reserved for high priority system processes. It is exempt from dispatcher decay and has a numeric range of 0 to 99.
- The BS Queue contains both the system and some carefully placed user processes. It is exempt from dispatcher decay and has a numeric range of 100-150 (default).
- The CS Queue is for interactive user processes. The range is 152-200(default). The range may be changed via :TUNE or :ALTWG.
- The DS Queue is for high-priority batch jobs. The range is 202 to 238(default). The range may be changed via :TUNE or :ALTWG.
- The ES Queue is for lower priority jobs. The range is 240 to 253(default). The range may be changed via :TUNE or :ALTWG.
- CS, DS, and ES Queues begin with a priority equal to the base and are decreased in priority until they reach the limit for the queue (circular queues).

## Queueing – Queue Rules

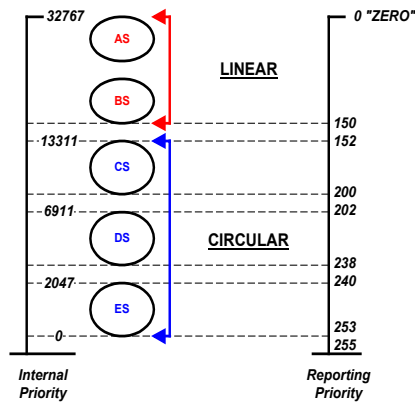
- SAQ: If a process uses more than its average, the priority is reduced (numerically increased.) This allows waiting processes a chance at some CPU time.
- CS, DS, and ES queues begin with a priority equal to the base and decrease in priority until they reach the limit for the queue (circular queues).
- Circular queue processes are lowered in priority when the process uses more than a certain amount of CPU time (dispatcher decay).
- High priority processes (AQ, BQ) are typically system processes and receive the CPU when needed.
- The Dispatcher penalizes long transactions by decreasing their priority and rewards short transactions.
- CPU does not wait on processes missing needed data, but will move on to the next process.
- Processes are assigned a characteristic of "exempt" or "non-exempt." "Exempt" do not have their priority decreased.

## Queueing – Dispatcher and Queueing



- Dispatcher
- Circular
- Base / Limit
- Decay / Boost
- Oscillate
- Preemption
- Timeslice
- Tune / Showq

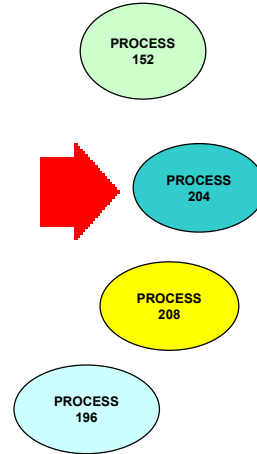
## Queueing – Internal Queueing and Preemption



- MPE/iX "or XL" = When MPE reports an internal process priority it translates it into the MPE/Classic "V" format
- AS queue processes can only be preempted by another AS Queue process with a higher priority
- Priorities decay when they exceed the SAQ or become subject to a Timeslice

## Queueing – Dispatcher and CPU

- The Dispatcher (ONLY 1) determines which process gets the system resources needed to run
- A process is defined as the unique execution of a particular program by a particular user at a particular time
- The Dispatcher favors processes that use the shortest processing time
- The Dispatcher makes sure that the highest priority process gets its fair share of the CPU
- The Dispatcher manages ready processes



## Queueing – Tune Command

### TUNE

Changes the scheduling characteristics of the scheduling queues. These characteristics include base and limit priorities, quantum bounds (min and max), boost property and timeslice. (NM)

### SYNTAX

```
{CQ}
TUNE [minclockcycle] [[:] {DQ}=[base], [[limit] [, [min] [, [max]
      {EQ}
[, [[decay   }]] [, [tslice]]]]]
  {oscillate}

[[:]...
```

**CAUTION** Misuse of this command can significantly degrade system operating efficiency

### Other Commands

ALTPROC – SHOWPROC - SHOWQ - SHOWWG - RUN

## Queueing – Tune Command

To set the CQ base to 152, limit to 200, and CQ maximum filter to 300; and the DQ base to 202, DQ limit to 238, DQ (and EQ) minimum and maximum filter to 1000, and cause oscillation boosting enter:

```
TUNE CQ=152,200,300,300;DQ=202,238,1000,1000,OSCILLATE
```

To set the CS queue to oscillation with a 300 millisecond timeslice and the DS queue's base to 180, limit to 238, timeslice to 1500 and boost property to decay enter:

```
TUNE=CQ=,,,OSCILLATE,300;DQ=180,238,,,DECAY,1500
```

## Queueing – ALTPROC Samples

- To alter process 133 and its current descendants so that their priorities execute within the DS\_Default workgroup, enter:
  - **ALTPROC #p133; TREE; wg=DS\_Default**
- To alter process 133 and its current descendants so that their scheduling queue attribute is DS, enter:
  - **ALTPROC #p133; tree; pri=DS**
- The outcome of this command is not necessarily identical to the outcome achieved with the previous command. If the system was configured with a user-defined workgroup that captured the processes. (MEMB\_QUEUE=DS and a match on other specified membership attributes as well), then the processes would be a member of the user-defined workgroup rather than the DS\_Default workgroup. To alter all job processes to the CS\_Default workgroup, enter:
  - **ALTPROC job=@j; wg=CS\_Default; anyuser1**

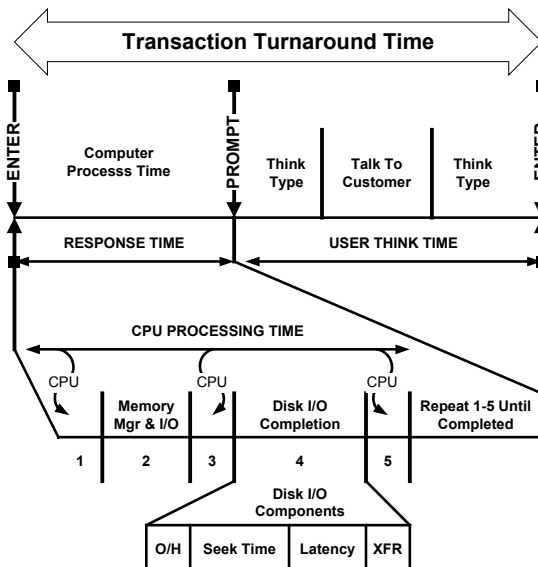
## Queueing – ALTPROC Samples

- To return the processes modified by the previous example to their natural workgroup(s), enter:
  - **ALTPROC job=@j; wg=NATURAL\_WG; anyuser**
- To alter all job processes matching the user's name to the CS\_Default workgroup, enter:
  - **ALTPROC job=@j; wg=CS\_Default; user**
- To alter the current process' priority so that it behaves like a CS queue manager (SM capability required), enter:
  - **ALTPROC 0;pri=CM**
- To alter all processes logged on as mgr.lund to linear 155 (SM capability required), enter:
  - **ALTPROC job=mgr.lund; pri=155**
- To alter the queue attributes of pins 150, 247, 211 to be ES, enter:
  - **ALTPROC (150,#p247,211); pri=ES**

## Queueing – Anatomy of a Process

- **The following page shows a real-time, interactive process**
- **Processes are more complex than most people think**





## Queueing – Three Things ...

Question? - When is a process launched?

Answer! - When "ALL" of the following have occurred:

- A Terminal read occurs
- The process's working set is resident
- It has the highest priority of all ready processes





## Queueing – Process Stop Reasons

- Preemption
- The SAQ - System Average Quantum
- Blocked on a resource
- Wait States - Wait States can be used to analyze why performance on a machine is not optimal
  - CPU Mem
  - Dsc Imp
  - Pre RIN
  - TWr BIO
  - Tim FS
  - Msg Oth
- Wait states are brick walls to process throughput. A brick wall could mean a missing memory segment, disk, or perhaps prevented access to a Turbo Image database.
- The most ideal throughput for a process is derived when it does not have to stop for any reason.



## Queueing – Process Preemption

