NOS 2.7.1 L716
FEATURE NOTES

SMD132216

Table of Contents

Introduction

NOS 2.7.1 L716 is the latest in Control Data Corporation's ongoing effort to provide efficient, reliable, and usable products to its customers. The NOS 2.7.1 L716 Feature Notes contain descriptions of the major enhancements included in the NOS 2.7.1 L716 release.

In general, Feature Notes are targeted for the site analyst. However, some of the topics covered may be of interest to site operations or the end user. We have organized this document such that each chapter may be easily copied and distributed. The Feature Note Audience Matrix (included in the laser copy only) is designed to serve as a guide for distribution of the articles contained in the Feature Notes.

These Feature Notes were jointly developed and written by CYBER Software Support and NOS Customer Support. Questions or comments regarding them may be addressed to:

Control Data Corporation
CYBER Software Support - ARH213
4201 North Lexington Avenue
Arden Hills, MN 55126-6198
USA

800-345-9903 (USA and Canada)
612-851-4131 (International)

Copies of the Software Release Bulletin and Feature Notes are available in line printer format on a separate permanent file tape shipped with the NOS 2.7.1 L716 release from Software Manufacturing and Distribution (SMD). A laser-printed hardcopy will also be included with NOS orders.

FEATURE NOTE AUDIENCE MATRIX

Article Title	Site/Analyst	Operations	End User
9853 Disk Subsystem Support	X	X	
NOS Scheduler Enhancements	X	X	

CHAPTER 1

9853 Disk Subsystem Support

NOS 2.7.1 L716 introduces support for the 9853 disk subsystem. Previously offered only for use with NOS/VE, the 9853 is a buffered disk device supported on NOS by the concurrent I/O (CIO) PP driver 1XD. The 9853 subsystem may be used with the CYBER 180-860, 180-960, 180-990, 180-994, and 180-995 mainframes configured with concurrent PPs.

The capacity of one standard 9853 disk in a NOS environment is 1151 megabytes. NOS utilizes the 2K sector size available on a 9853. To illustrate a measure of the 9853's transfer capabilities, the sustained transfer rate on a 180-860 mainframe for one 9853 is 3,120,000 characters per second.

(For an overview of the terminology and hardware components of the CIO system, reference the NOS 2.5.1 L664 Feature Notes, publication number SMD110168.)

1.1 Configuration Specifications

1.1.1 Hardware and Maintenance

A maximum of 80 dual access drives may be configured on one mainframe. Though not recommended for fault tolerance reasons, up to 160 single access drives can be configured on one mainframe. As previously indicated, the 9853 disk subsystem driver requires that the system be configured with an I4 IOU and CIO subsystem; the 9853 is accessed via IPI DMA channels.

The primary maintenance strategy is to utilize in-line diagnostics supplied with the subsystem. All error information concerning the IPI DMA channel and the 9853 subsystem is logged in the Binary Maintenance Log (BML). As with other NOS disks, data path verification is performed after a level 0 deadstart by 1MV before users can access the system.

1.1.2 Restrictions

Because a CIP or disk deadstart requires an NIO channel, the 9853 does not support CIP residency or disk deadstart of the operating system. This means that a disk subsystem other than the 9853 must be configured on the mainframe for use as a CIP device and/or a deadstart device. The 9853 can coexist with the 7x5x/844, 7155/844-4x, 7155/885-1x and 7165/895 disk subsystems (all of which support deadstart) as well as the 887 disk subsystem (which does not support deadstart).

9853 disks may not be shared (ISHARE or MMF) between mainframes but may coexist with other disk subsystems that are shared between mainframes.

1.2 New Deadstart Deck Entries

1.2.1 EQPDECK Entry for the 9853

The EQPDECK entry for the 9853 subsystem is:

```
EQest=DN,ST=stat,EQ=eq,UN=un,CH=ch1pt1/ch2pt2,AP=ap,IB=ib.
```

Parameter	Description
est	EST ordinal of the disk units; from 5 to 777B.
stat	Specifies status indicating whether or not the equipment is available for access; enter one of the following values:

Status	Description
DOWN	All access to the equipment is inhibited.
IDLE	No new files can be assigned to the device.
OFF	No user jobs can access the equipment.
ON	The equipment is generally available.

eq	Control module number for the 9853 disk; from 0 to 7B.
un	Unit numbers; from 0 to 7B.
ch1	Channel number for primary access; from 0 to 11B.
pt1	Channel port (A or B) for the primary access channel. Default is port A.
/ch2	Channel number for secondary access; from 0 to 11B. The primary and secondary access channel numbers must be different.

pt2 Channel port (A or B) for the secondary access channel. Default is port A.

ap The 1- or 2-digit octal number that indicates which APRDECK to use. If AP is omitted, APRD00 is assumed.

ib The optional 1- to 4-digit octal value entered in the installation specified EST ordinal.

1.2.2 Using a 9853 Disk Subsystem After Use by NOS/VE

If the 9853 subsystem has been previously used by NOS/VE, the control module may be left in a state that renders it unrecognizable to NOS. When this occurs, the new command

```
ENABLE,CM RESET.
```

may be entered in the IPRDECK or at the system console to clear the control module prior to disk initialization.

NOTE

The process of resetting a control module takes an extended period of time. Up to five minutes per control module will be required.

1.3 Affected Documentation

The following manuals have been updated for the NOS 2.7.1 L716 release to include information about the 9853 disk subsystem.

NOS Version 2 Analysis Handbook, publication no. 60459300
NOS Version 2 Operations Handbook, publication no. 60459310
NOS Version 2 System Programmer's Instant, publication no. 60459370
NOS Version 2 Reference Set, Volume 3, publication no. 60459680
NOS Version 2 BML Message Handbook, publication no. 60459940
NOS Version 2 Installation Handbook, publication no. 60459320

CHAPTER 2

NOS Scheduler Enhancements

NOS 2.7.1 L716 includes several enhancements to the NOS scheduler. The objectives of this feature were as follows:

- Improve Central Memory Utilization. Increase the maximum number of jobs that can be contained in central memory at one time. This alleviates the control point contention on many large memory mainframes.

- Reduce Storage Moves. Rework the method of allocating central memory and extended memory, to reduce the need to move jobs in order to allocate memory.

- Provide CPU Scheduling Flexibility. Allow a site to give preference in CPU scheduling to jobs of one service class over fhose in another class, without having the preferred class totally dominate the CPU.

The implementation of the NOS scheduler enhancements has been designed in such a way that sites which prefer to achieve the same scheduling environment as that of older levels of NOS may continue to do so with minimal effort. Refer to the CPU Scheduling Parameter Selection topic in the Scheduler Tuning section for the required IPRDECK changes.

2.1 Glossary

2.1.1 Compound rollout

A compound rollout is the operation involving the rollout of a pseudo-control point job to mass storage, followed by a pseudo-rollout of another job from a control point to the vacated pseudo-control point.

2.1.2 CPU Switch Delay

The CPU switch delay specifies the number of milliseconds that a job can keep the CPU once it has been assigned. Until the switch delay has expired, the job can not lose the CPU even if other jobs with higher CPU priorities are waiting for the CPU.

2.1.3 Memory Control Table (MCT)

The MCT is a central memory resident table used to identify the order of RAs and RAEs relative to control points and psuedo-control points. (RA, or reference address, refers to the absolute central memory starting address; RAE is the absolute extended memory starting address). This table consists of two word entries, one entry per control point or pseudo-control point, indexed by the control point/psuedo-control point number. Word MCTP in CMR points to the first word of the MCT. Control point 0 always has the lowest RA/RAE (zero); therefore, it is first in the chain of MCT entries. Control point 1 follows control point 0. The MCT entries for pseudo-control points follow the MCT entry for the system control point; however, the last part of central memory is always allocated to the last control point, not to the last PCP. The MCT entries link forward and backward for ease of management.

2.1.4 Pseudo-Control Point (PCP)

The pseudo-control point is a mechanism to allow a job to physically remain in central memory while it is logically rolled out. This is an extension of the existing NOS control point (CP) concept. Unlike a job at a control point, a job at a PCP cannot have activity of any kind (PP assignment, CPU assignment, etc.). Each PCP has an associated pseudo-control point area (PCPA) which is identical in structure to a control point area (CPA). Unlike CPAs, PCPAs are not required to reside below CM address 10000B. PCPs are numbered to make them logically contiguous with CPs. For example, in a system having N control points and K pseudo-control points, the CPs are numbered 1 through N, the system control point is N+1, and the PCPs are numbered N+2 through N+K+1.

2.1.5 Pseudo-rollin

A pseudo-rollin is the rollin of a job at a PCP to a control point.

2.1.6 Pseudo-rollout

A pseudo-rollout is the rollout of a job at a control point to a PCP.

2.1.7 Simple PCP rollout

A simple PCP rollout is the rollout of a job at a PCP to mass storage, leaving the PCP vacant.

2.1.8 Simple pseudo-rollout

A simple pseudo-rollout is a pseudo-rollout of a job to a vacant PCP, so that no PCP rollout is required prior to the pseudo-rollout.

2.2 Design

2.2.1 Central Memory Utilization Improvements

Improvements in central memory utilization were achieved by introducing the concept of a pseudo-control point (PCP). A job at a PCP is functionally rolled out, but is still physically present in central memory. When PCPs are defined, some jobs which would otherwise be rolled out to mass storage to free up a control point instead remain in memory and are pseudo-rolled to a PCP. Since the act of moving a job from a control point to a PCP and back again does not involve moving the job's field length, the overhead of rollout/rollin is greatly reduced.

2.2.2 Storage Move Reduction

A reduction in storage moves was achieved by restructuring the allocation scheme for central memory and extended memory. In the past, memory was allocated sequentially; the memory for each control point was allocated following the memory of the previous control point. In NOS 2.7.1 L716, the new memory control table (MCT) allows memory to be allocated for control points and pseudo-control points in any order. The new allocation scheme allows memory for a new job to be allocated anywhere in memory where there is sufficient space.

When a job does have to move, it can frequently move directly to the destination memory space, without having to move other jobs that were merely in the way due to ascending control point/memory requirements.

In addition, the system now attempts to leave space between jobs in memory. This allows the system to satisfy many requests for additional memory without having to do a storage move.

2.2.3 CPU Scheduling Flexibility

Two tools were provided to enhance CPU scheduling flexibility: variable CPU switch delays and ranges of CPU priorities.

A different CPU switch delay may be selected for each service class. Selecting a larger CPU switch delay for a given service class results in jobs of that class receiving proportionally more CPU.

If a range of CPU priorities is defined for a service class, the CPU priority for a job in that service class is adjusted up and down within this range. By defining overlapping priority ranges for different service classes, it is possible to give preference to one service class without allowing it to monopolize the CPU.

2.3 DSD and Deadstart Deck Changes

2.3.1 New CMRDECK Entry PCP

The new CMRDECK entry PCP specifies the number of pseudo-control points to be defined. The format for this entry is as follows:

```
PCP=nn.
```

The value for nn can range from 0 to 34 octal. If the PCP entry is not specified in CMRDECK, no pseudo-control points are defined.

2.3.2 Changes to IPRDECK and DSD DELAY Command

The MX and MN parameters have been removed from the DELAY command. The CPU switch delay is now defined at the service class level using the SERVICE command SD parameter.

Two new parameters, CI and MP, have been added to the DELAY command. CI specifies the frequency of incrementing the CPU priority of jobs waiting for the CPU (W status) in milliseconds. MP specifies the memory padding factor used when processing central or extended memory allocation. MP is expressed in allocation blocks (100B word blocks for central memory and UEBS size blocks for extended memory, where UEBS is a factor determined by the size of available extended memory).

2.3.3 Changes to IPRDECK and DSD SERVICE Command

An SD parameter has been added to the SERVICE command to allow specification of the CPU switch delay (previously defined for all jobs by the DELAY command MX and MN parameters) at the service class level. SD is expressed in milliseconds.

The PR paramater has been replaced by the new CB parameter on the SERVICE command to allow specification of a range of CPU priorities for each service class. CB is expressed as four octal digits in the form CBlpup, where lp is the CPU priority lower bound and up is the CPU priority upper bound. Setting lp and up to the same value defines a constant CPU priority for the service class, which is equivalent to the old PR parameter. If a range of CPU priorities is defined for a service class, the CPU priority for a job in that service class is adjusted up and down within this range, based on the values specified on the SERVICE command SD parameter and the DELAY command CI parameter. The CPU priority starts at the upper bound when the job is rolled in from mass storage. The priority is decremented by one every time that the job exceeds the CPU switch delay (as defined by the SERVICE command SD parameter), unless the CPU priority is above 57B or is already at the lower bound. For each job waiting for the CPU, the CPU priority is incremented by one each time that the interval specified by the DELAY command CI parameter expires, unless the priority is already at the upper bound.

The meaning of the SERVICE command CP parameter has been changed. Previously, CP defined the CPU time slice, which was used in control point scheduling. The CPU time slice has been eliminated from the current system. CP now defines a control point time slice scheduling priority. An executing job's scheduling priority is decreased to this value when the control point time slice defined by the CT parameter expires. The CP and CT parameters are used in scheduling jobs between control points and pseudo-control points.

A CT parameter has been added to the SERVICE command; CT defines a job's control point time slice. When an executing job has exceeded its control point slice, the scheduling priority is set to the control point slice priority defined by the CP parameter. This makes the job a likely candidate for rollout to a pseudo-control point.

NOTE

The PR parameter is no longer allowed on the SERVICE command. Sites upgrading from previous NOS levels will need to modify their IPRDECK to reflect this change or adapt an example IPRDECK provided with the release materials.

2.3.4 New DSD Command ENPR

The following new DSD command may be used to change a job's CPU priority:

```
ENPR,jsn,pr.
```

Values of 70B and above may not be specified for pr. If a priority of * is specified, the job's CPU priority is set to 60B (the lowest non-decrementing CPU priority).

2.3.5 Changes to DIS Command ENPR

Values of 70B and above may no longer be specified on the DIS command ENPR. If a priority of * is specified, the job's CPU priority is set to 60B (the lowest non-decrementing CPU priority).

2.3.6 Changes to the SDSPLAY L Display Utility

The SDSPLAY L Display utility has been changed to reflect the new and modified parameters of the SERVICE command. This gives the operator an alternate method for changing the values of scheduling parameters. The CB, CT and SD keywords are added; the CP keyword is redefined; the PR keyword is no longer valid.

2.3.7 Changes to DSD Displays

The R display now displays a job status of "PC" for any job at a pseudo-control point.

The S display now displays the values of CB, CP, CT and SD for each service class.

The W,M display now displays the value of the CI parameter specified on the DELAY command.

The W,P display is added at this release to show first word addresses of many system tables including the MCT.

The W,R display now displays the number of free pseudo-control points, if PCPs are present in the system.

2.4 Scheduler Tuning

In order to gain the greatest advantage from this enhancement, various system parameter values must be properly set.

2.4.1 Memory Pad Selection

The DELAY command MP parameter defines the memory pad value. The purpose of the memory pad is to keep jobs separated from one another in central memory. When a job rolls in or is storage moved, a space equal to the memory pad value is left in between that job and the following job. If the job subsequently asks for additional field length, it is then possible to satisfy the request without having to perform a storage move.

The strategy of pad selection should be to choose the smallest pad large enough to accommodate a typical memory increase, or set of increases. This value may be highly dependent on your particular workload.

Care should be taken not to set the pad size too large, since this may cause excessive memory fragmentation. Such fragmentation could be detrimental to performance in an environment containing jobs of widely varying field lengths.

2.4.2 Pseudo-Control Point Usage

When defined in certain types of system environments, pseudo-control points can improve system performance. However, since there are situations in which PCP usage can actually degrade system performance, it is important to understand when and when not to use them.

PCPs should be thought of as providing a solution to a specific problem: excessive rollin/rollout activity caused by control point contention. A site not experiencing a control point contention problem is unlikely to benefit from PCP usage. Control point contention tends to be a malady most often seen in mainframes having a large CM size, say in excess of a half million words.

The presence of PCPs in a configuration without control point contention may cause increased storage move activity, reducing CPU availability for user job processing.

To illustrate a case where PCP usage would be beneficial, consider a system environment having the following characteristics:

- Enough CM so that CM saturation is a rare occurrence.
- 20 PPs.
- Buffered disks (885-42, 887, 895 or 9853 disks).
- Several subsystems active, reducing the number of available control points.
- A batch and interactive workload mix, with interactive jobs generally running at higher scheduling priorities than batch jobs.

In an environment like this, batch jobs are frequently rolled out to free up control points for usage by interactive jobs. If an interactive job then rolls out for terminal I/O after a relatively short time, the batch job that rolled out a few seconds (or less) ago may now be rescheduled and roll back into the vacant control point. Since the rollout and rollin of a job is a very costly process, the use of PCPs in this case may significantly reduce system overhead.

2.4.2.1 Selecting the Number of PCPs

Care should be taken to avoid defining too many PCPs. First, each PCP requires 200B words of central memory and an MCT entry. Additionally, as more PCPs are added, more jobs are held in CM at one time. This may lead to CM saturation, causing increased storage move overhead.

A reasonable approach to selecting the number of PCPs is to start with the number of control points minus the number of dedicated control points (those occupied by subsystems). Using PROBE and/or ACPD output as a guide, lower the number of PCPs until CM saturation no longer occurs during peak workload periods.

2.4.2.2 Setting the CT and CP SERVICE Parameters

The SERVICE command CT and CP parameters affect pseudo-control point management. The CT parameter defines the control point time slice; it specifies how many seconds a job may reside at a control point before its scheduling priority is reduced to the control point slice priority, CP. When CT and CP parameters are properly selected, jobs in central memory are circulated between control points and pseudo-control points in an orderly fashion, in accordance with the relative priorities of their service classes. If no pseudo-control points are present, the CT and CP parameters are meaningless.

When a job is at a pseudo-control point, its scheduling priority is incremented periodically just as if it were rolled out on disk. Because of this, a good starting point for selecting CP and CT values is to set CP to some value close to, but less than the upper bound (UP) and set CT to UP-CP. For example, if UP=7000, set CP to 6770 and set CT to 10. CT should be set lower than CM (central memory time slice).

2.4.2.3 Miscellaneous PCP Considerations

There may be some advantage to reducing the number of control points and increasing the number of pseudo-control points. There is significant overhead in managing the CPU wait queue and the magnitude of this overhead is proportional to the length of the queue. The length of the queue is limited by the number of control points.

When PCPs are defined to relieve the batch job rollin/rollout thrashing characteristics of some batch and interactive job mixes, it may be possible to boost interactive response time with relatively little effect on batch throughput. This can be done by setting the IP and TP SERVICE parameters for the interactive service class to the upper bound scheduling priority for the batch service class.

When PCPs are present, it is desirable to assign a subsystem to the first control point and to the last control point to ensure full control point utilization during times of heavy PCP usage. The structure of the MCT requires that the first portion of memory always be allocated to control point one and the last portion of central memory always be allocated to the last control point. Because of this, a job at a PCP cannot be scheduled to either of these control points; conversely, a job stationed at either of these control points cannot be pseudo-rolled to a PCP.

2.4.3 CPU Scheduling

To achieve greater flexibility in the area of CPU scheduling, a two part strategy has been used. First, different CPU switching delays may now be selected for each service class. Second, a CPU priority bounding approach has been implemented.

2.4.3.1 CPU Switch Delay

The SD parameter on the SERVICE command specifies the CPU switch delay for a job of that service class. The switch delay specifies the number of milliseconds that a job can keep the CPU once it has been assigned. Until the switch delay has expired, the job cannot lose the CPU even if other jobs with higher CPU priorities are waiting for the CPU. This parameter can be used to favor one service class over another in CPU allocation.

Consider the following example. There are two competing service classes, BC (batch) and TS (interactive). Suppose the CPU priority bounds for these two service classes are identical, but the switch delay for service class BC is 40 while the switch delay for service class TS is 20. If only one job of each service class is executing and both jobs are CPU bound rather than I/O bound, the interactive job receives 1/3 of the CPU while the batch job receives 2/3.

Now, suppose two interactive jobs are running with only one batch job, again with all jobs being CPU bound rather than I/O bound. The interactive jobs each receive 1/4 of the CPU while the batch job receives 1/2. So, varying the number of jobs in the service classes affects the total CPU allocation relative to the service class but the ratio of CPU allocation of a job in one service class to a job in another service class remains constant. In the present example, the amount of CPU allocated to any interactive job is always one half of the amount allocated to any batch job during a given time interval of sufficient length.

There are advantages and disadvantages to larger switch delays. CPU switching is expensive in terms of system monitor mode CPU usage. Excessive CPU switching caused by inadequate switch delays may have detrimental performance consequences; larger switch delays yield fewer CPU switches. However, large switch delays can allow some I/O bound jobs to dominate the CPU, particularly on systems with buffered disks. In addition, very large switch delays may hurt interactive response time, if an interactive job with a high priority must wait a long time to get the CPU from a competing job because of a large switch delay for the other job's service class.

2.4.3.2 CPU Priority Ranges

The SERVICE command CB parameter allows the selection of a range of CPU priorities for each service class. This parameter can be used to favor one service class over another in CPU scheduling, but the effect is somewhat less predictable than that of the SD parameter.

To illustrate the considerations underlying the selection of the CPU priority bounds for a particular service class, consider the following example. Service class BC (batch) has a CPU priority range of 30-32 (CB3032), service class TS (interactive) has a range of 30-34 (CB3034), and service class MA (maintenance) has a fixed CPU priority of 2 (CB0202). In this environment, if three CPU intensive jobs were simultaneously executing, with one job being from each of the three service classes, the CPU would be assigned to the interactive job five eighths of the time, the batch job three eighths of the time, and the maintenance job would never get the CPU. This assumes negligible system CPU overhead during the interval in which the measurement is taken and that the switch delays selected for the BC and TS service classes are identical.

In an actual production environment, even if we restrict the discussion to only the BC and TS service classes, the scheduling situation would be more complex. For example, if there were five interactive jobs and only two batch jobs executing, CPU usage would be skewed heavily in the direction of the TS service class. The degree of weighting CPU allocation toward one service class or another is a function of several factors including:

1. The CPU priority bounds of each competing service class.
2. The switch delays of each competing service class.
3. The CPU priority incrementing delay specified on the DELAY command.
4. The number of jobs executing in each competing service class.
5. The amount of I/O performed by the jobs.

In this context, competing service classes refers to service classes of jobs that are intended to actively share the CPU. In the present example, the MA service class is not a competing service class. Maintenance jobs run as background jobs and never compete with batch or interactive jobs for the CPU.

2.4.3.3 CPU Priority Incrementing Delay

The DELAY command CI parameter may be used to age the CPU priority upward for jobs in the CPU wait queue. This can be useful for certain combinations of CPU priority ranges.

Consider the following example. There are two competing service classes, BC and TS. The CPU priority range for service class BC is 26-32 (CB2632) and the range for service class TS is 30-34 (CB3034). In this example, the batch and interactive priority ranges overlap but the lower bounds are unequal. Because of this, a batch job whose priority has been decremented to 27 would be excluded from using the CPU whenever any interactive job is executing.

If this is not the desired effect, the DELAY command CI parameter may be used to cause jobs in the CPU wait queue to be aged upward toward their upper bound CPU priorities. This process tends to offset the downward aging process which occurs whenever a job exceeds the CPU switch delay. Since CI aging is performed regardless of whether the job is getting the CPU, the value specified for CI should be significantly larger than the values specified for SD.

2.4.3.4 CPU Scheduling Parameter Selection

The selection of CPU scheduling parameters should be approached with caution; an inappropriate selection can cause severe system performance degradation.

In previous versions of NOS, it was very difficult to favor one job, or class of jobs, without either excluding all other jobs from execution or introducing excess system overhead such as additional rollout/rollin activity. While it was possible to set different CPU priorities for different service classes, it usually did not work very well. This was because the CPU could not be actively shared by jobs having differing CPU priorities. Only when the higher priority job was rolled out or did not need the CPU could the lower priority job get the CPU. The new CPU scheduling parameters provide a means to solve this problem.

For the site satisfied that the old NOS CPU scheduling method meets their needs, the new parameters should be set so as to preserve the execution environment yielded by previous versions of NOS. This can be done by setting the upper and lower bounds equal to each other for all service classes. Furthermore, the upper and lower bound values of competing service classes should be identical from one service class to the next. For example, the site could select CB3030 for each competing service class. Switch delays for all service classes could be set to 20 and the CPU priority incrementing delay, CI, could be set to 7777.

If a site wishes to favor a particular service class in terms of CPU allocation, the new parameters provide a variety of methods; however, how well these methods work is highly workload dependent. The PROBE and TRACER utilities have been updated to report data relative to the new scheduling modifications; they should be used to evaluate the effects of various parameter changes.

One method of favoring a service class would be to increase the switch delay for the favored service class, while setting the upper and lower CPU priority bounds for all competing service classes to a constant value (for example, CB3030).

Another method of favoring a service class, or in broader terms, establishing a priority order among competing service classes, is to define overlapping CPU priority ranges that achieve specific throughput objectives. The simplest starting point would be to set all lower bounds of competing service classes to the same value, while varying the upper bounds. Having identical lower bounds has the advantage of eliminating the need to age the wait queue; this allows you to set CI to 7777, to minimize the system overhead involved in aging the CPU wait queue.

In the final analysis, it will be necessary for a site to experiment with CPU switch delays and/or CPU priority bounds to achieve a particular distribution of the CPU(s) across competing service classes.

2.4.3.5 Dual State Considerations

When selecting CPU scheduling parameters for either NOS or NOS/VE in a dual state environment, the site must maintain a more global view of CPU allocation. Higher CPU priorities given to jobs of one system reduces the system performance on the other system.

In a dual state environment, caution should be exercised over the use of CPU priority ranges for user jobs. There are two reasons for this:

1. The NVE subsystem, which runs on the NOS side as a normal job, forces its CPU priority to 30. If user jobs are running above this range, the NVE subsystem may not be able to get enough CPU. This could cause problems for NOS/VE, particularly during initiation and termination.

2. The dual state CPU allocation algorithm bases its decisions on the upper digit of the NOS CPU priority. NOS jobs with CPU priorities in the range of 30-37 are considered to be normal jobs, and can compete with NOS/VE user jobs on a somewhat equal basis (subject to NOS/VE scheduling parameters). NOS jobs with CPU priorities of 27 or below have a significant disadvantage in competing with NOS/VE jobs; NOS jobs with CPU priorities of 40 or above have a significant advantage. Therefore, if CPU priority ranges are specified for user jobs, they should probably be in the range of 30-37.

2.5 Affected Documentation

The following manuals have been updated for the NOS 2.7.1 L716 release to include information about the NOS scheduler enhancements. Of particular interest will be the NOS Analysis Handbook and the System Programmer's Instant.

NOS Version 2 Analysis Handbook, publication no. 60459300
NOS Version 2 Operations Handbook, publication no. 60459310
NOS Version 2 System Programmer's Instant, publication no. 60459370
NOS Version 2 Installation Handbook, publication no. 60459320