

ASR1000 시리즈 라우터의 높은 CPU 문제 해결

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[설명](#)

[문제 해결 단계](#)

[1단계 - CPU가 높은 모듈 식별](#)

[2단계 - 모듈 분석](#)

[3단계 - IOS 프로세스](#)

[4단계 - Linux 프로세스](#)

[5단계 - FECF 프로세스](#)

[6단계 - QFP 사용률](#)

[7단계 - 근본 원인 파악 및 해결 방법 확인](#)

[트러블슈팅 예](#)

[추가 명령](#)

[Route Processor](#)

[Embedded Services Processor](#)

소개

이 문서에서는 ASR1000 Series 라우터의 높은 CPU 문제를 해결하는 방법에 대해 설명합니다.

사전 요구 사항

요구 사항

이 문서를 해석하고 활용하려면 [ASR1000 아키텍처](#)를 이해하는 것이 좋습니다.

설명

Cisco 라우터의 높은 CPU는 라우터의 CPU 사용률이 정상 사용량을 초과하는 조건으로 정의할 수 있습니다. 일부 시나리오에서는 CPU 사용률이 증가할 것으로 예상되지만, 다른 시나리오에서는 문제가 발생할 수 있습니다. 네트워크 변경 또는 컨피그레이션 변경으로 인한 라우터의 일시적인 높은 CPU 사용률은 무시될 수 있으며 이는 정상적인 동작입니다.

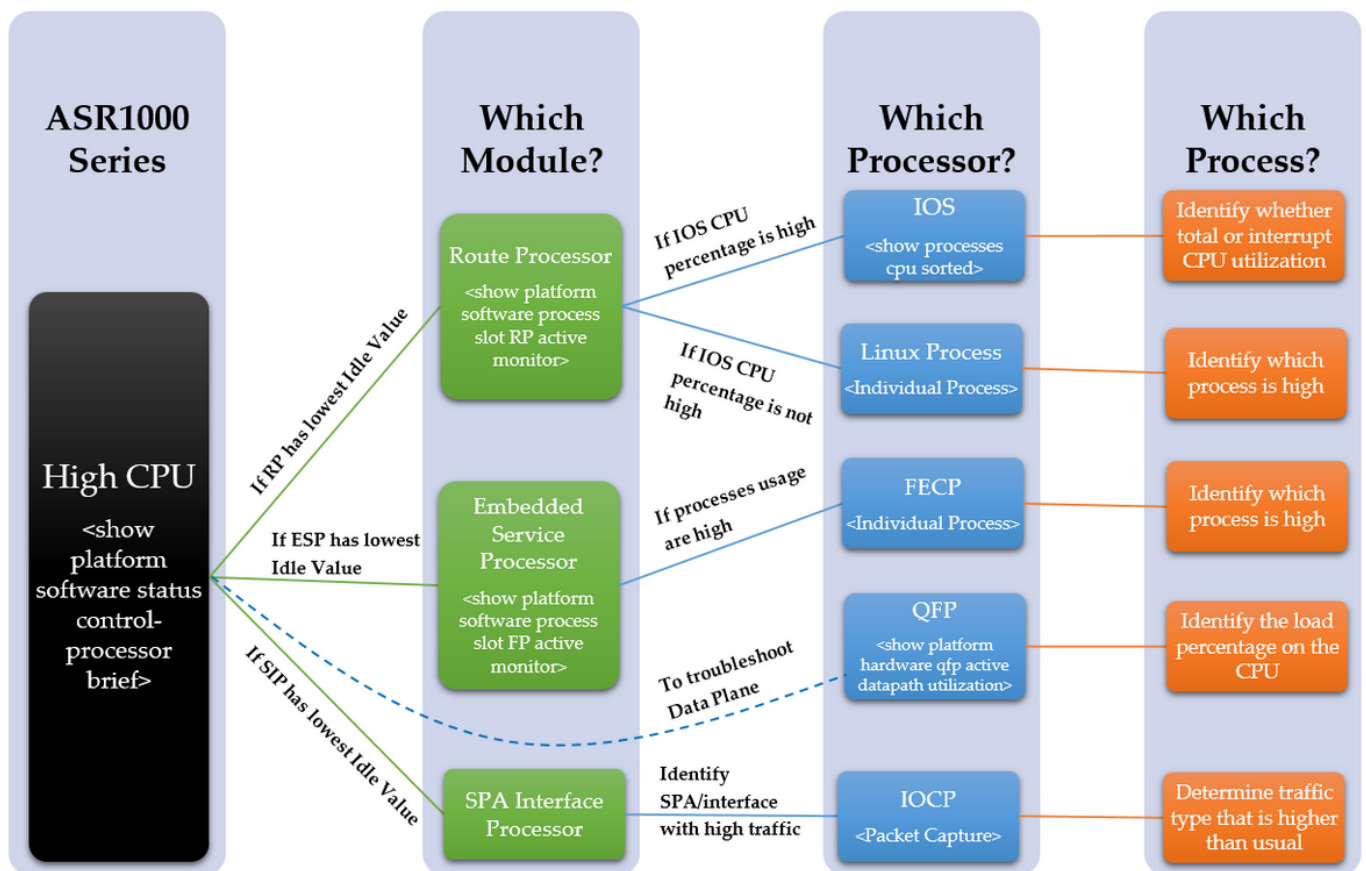
그러나 네트워크나 구성을 변경하지 않고 장기간 CPU 사용률이 높은 라우터는 이례적인 것으로 분석해야 합니다. 따라서 CPU가 너무 많이 사용되면 다른 모든 프로세스에 능동적으로 서비스를 제공하지 못하기 때문에 명령줄, 컨트롤 플레인 지연, 패킷 삭제 및 서비스 실패가 느려집니다.

높은 CPU의 원인은 다음과 같습니다.

1. 컨트롤 플레인 CPU가 너무 많은 punt 트래픽을 수신합니다.
2. 예기치 않게 작동하여 CPU 초과 사용률이 발생하는 프로세스
3. 데이터 플레인 프로세서가 과도하게 사용/초과 서브스크립션
4. 프로세서 인터럽트가 너무 많습니다.

라우터 CPU 사용률이 라우터의 부하에 정비례하기 때문에 CPU가 높다고 항상 ASR1000 Series 라우터 문제가 되는 것은 아닙니다. 예를 들어, 네트워크 변경이 있는 경우 네트워크가 다시 통합되므로 제어 평면 트래픽이 대량으로 발생합니다. 따라서 CPU 초과 사용률의 근본 원인을 파악하여 예상된 동작인지 또는 문제인지 확인해야 합니다.

다음은 높은 CPU 문제를 해결하는 방법에 대한 단계별 프로세스를 자세히 설명하는 다이어그램입니다.



문제 해결 단계

1단계 - CPU가 높은 모듈 식별

ASR1000은 서로 다른 모듈에 여러 개의 서로 다른 CPU가 있습니다. 따라서 어떤 모듈이 정상 사용량보다 더 높게 나타나는지 확인해야 합니다. 이는 Idle(유휴) 값을 통해 알 수 있으며, 유휴 값이 낮을수록 해당 모듈의 CPU 사용률이 높아집니다. 이러한 다양한 CPU는 모두 모듈의 컨트롤 플레인을 반영합니다.

디바이스 내에서 어떤 모듈이 높은 CPU를 경험하는지 확인합니다. 아래 명령을 사용하는 RP, ESP 또는 SIP입니까?

show platform software status control-processor 개요

강조 표시된 열을 보려면 아래 출력을 참조하십시오

RP의 유휴 값이 낮은 경우 2단계 지점 1로 진행합니다

ESP의 Idle(유휴) 값이 낮은 경우 3단계 포인트 2로 진행합니다.

SIP의 유휴 값이 낮은 경우 4단계, 3단계로 진행합니다

Router#**show platform software status control-processor** 개요

로드 평균

슬롯 상태 1분 5분 15분

RP0 정상 0.00 0.02 0.00

ESP0 정상 0.01 0.02 0.00

SIP0 정상 0.00 0.01 0.00

메모리(kB)

슬롯 상태 총 사용됨(Pct) 사용 가능(Pct) 커밋됨(Pct)

RP0 정상 2009376 1879196 (94%) 130180 (6%) 1432748 (71%)

ESP0 정상 2009400 692100 (34%) 1317300 (66%) 472536 (24%)

SIP0 정상 471804 284424 (60%) 187380 (40%) 193148 (41%)

CPU 사용률

슬롯 CPU 사용자 시스템 Nice 유휴 IRQ SIRQ IOwait

RP0 0 2.59 2.49 0.00 **94.80** 0.00 0.09 0.00

ESP0 0 2.30 17.90 0.00 **79.80** 0.00 0.00 0.00

SIP0 0 1.29 4.19 0.00 **94.41** 0.09 0.00 0.00

Idle 값들이 모두 상대적으로 높다면, 이는 제어 평면 문제가 아닐 수 있다. 데이터 플레인의 문제를 해결하려면 ESP의 QFP를 준수해야 합니다. 과도하게 사용된 QFP로 인해 여전히 "CPU 사용량이 많은" 증상이 관찰될 수 있으며, 이는 컨트롤 플레인 프로세서에서 CPU 사용량이 많은 결과를 초래하지 않습니다. 6단계로 진행합니다.

2단계 - 모듈 분석

- Route Processor

아래 명령을 사용하여 RP에서 어떤 프로세서가 CPU 사용률이 높은 것으로 관찰되었는지 확인합니다. Linux 프로세스입니까, IOS 프로세스입니까?

플랫폼 소프트웨어 프로세스 슬롯 RP 활성 모니터 표시

IOS CPU 백분율이 높으면(linux_iosd-imag) RP IOS가 됩니다. 3단계로 진행합니다

다른 프로세스의 CPU 비율이 높으면 Linux 프로세스일 가능성이 높습니다. 4단계로 진행합니다

- Embedded Services Processor

ESP에서 컨트롤 플레인 프로세서의 CPU 사용률이 높은지 확인합니다. 배설물인가요?

플랫폼 소프트웨어 프로세스 슬롯 FP 액티브 모니터 표시

공정이 많을 경우 FECF이며, 5단계로 진행합니다

FECF가 아닌 경우 ESP 내에서 관련된 문제를 처리하는 컨트롤 플레인이 아닙니다. 네트워크 지연 또는 대기열 삭제와 같은 증상이 여전히 관찰되는 경우 데이터 플레인의 초과 사용을 검토해야 할 수 있습니다. 6단계로 진행합니다

- SPA Interface Processor

SIP의 CPU 사용률이 높은 경우 IOCP의 CPU 사용률이 높게 나타납니다. IOCP 내에서 CPU 사용률이 높은 것으로 관찰되는 프로세스 또는 프로세스를 결정합니다.

패킷 캡처를 수행하고 어떤 트래픽이 평소보다 높은지, 어떤 프로세스가 이 유형의 트래픽과 연결되었는지 확인합니다. 7단계로 진행합니다

3단계 - IOS 프로세스

아래 출력을 참조하십시오. 첫 번째 백분율은 총 CPU 사용률이고, 두 번째 백분율은 버스트 패킷을 처리하는 데 사용된 CPU 양인 인터럽트 CPU 사용률입니다.

인터럽트 백분율이 높으면 많은 양의 트래픽이 RP에 할당됨을 의미합니다(show platform software infrastructure punt 명령을 사용하여 확인할 수 있음)

인터럽트 백분율은 낮지만 총 CPU가 높은 경우, CPU를 장기간 사용하는 프로세스가 관찰됩니다.

IOS에서 아래 명령을 사용하여 CPU 사용률이 높은 것으로 관찰된 프로세스 또는 프로세스를 확인합니다.

```
show processes cpu sorted
```

높은 비율(총 CPU 또는 인터럽트 CPU)을 확인한 다음 필요한 경우 개별 프로세스/프로세스를 식별합니다. 7단계로 진행합니다

Router#**정렬된 프로세스 표시**

5초 동안의 CPU 사용률: 0%/0%; 1분: 1%; 5분: 1%

```
PID Runtime(ms) Invoked uSec 5Sec 1Min 5Min TTY Process
```

```
PID Runtime(ms) Invoked uSec 5Sec 1Min 5Min TTY Process
```

```
188 8143 434758 18 0.15% 0.18% 0.19% 0 Ethernet Msec Ti
```

```
515 380 7050 53 0.07% 0.00% 0.00% 0 SBC 주 프로세스
```

```
3 2154 215 10018 0.07% 0.00% 0.19% 0 Exec
```

```

380 1783 55002 32 0.07% 0.06% 0.06% 0 MMA DB 타이머
63 3132 11143 281 0.07% 0.07% 0.07% 0 IOSD ipc 작업
5 1 2 500 0.00% 0.00% 0.00% 0 IPC ISSU Dispatc
6 19 12 1583 0.00% 0.00% 0.00% 0 RF 슬레이브 Main Th
8 0 1 0 0.00% 0.00% 0.00% 0 R0 알림 타이머
7 0 1 0 0.00% 0.00% 0.00% 0 EDDRI_MAIN
10 6 75 80 0.00% 0.00% 0.00% 0 풀 관리자
9 5671 538 10540 0.00% 0.14% 0.12% 0 체크힙

```

4단계 - Linux 프로세스

IOS에서 CPU를 과도하게 사용하는 것으로 관찰될 경우 개별 linux 프로세스에 대한 CPU 사용률을 관찰해야 합니다. 이러한 프로세스는 show platform software process slot RP active monitor에서 나열한 다른 프로세스입니다. 높은 CPU를 경험하는 것으로 관찰되는 프로세스를 식별한 다음 7단계로 진행합니다.

5단계 - FECF 프로세스

프로세스 또는 프로세스가 높으면 CPU 사용률이 높은 FECF 내의 프로세스일 가능성이 높습니다. 7단계로 진행합니다

6단계 - QFP 사용률

Quantum Flow Processor는 포워딩 ASIC입니다. 포워딩 엔진의 로드를 확인하기 위해 QFP를 모니터링할 수 있습니다. below 명령은 입력 및 출력 패킷(우선순위 및 비우선순위)을 초당 패킷 수 및 초당 비트 수로 나열합니다. 마지막 줄에는 패킷 전달로 인한 총 CPU 로드와 백분율로 표시됩니다.

플랫폼 하드웨어 qfp 활성 데이터 경로 사용률 표시

입력 또는 출력이 높은지 확인하고 프로세스 로드를 확인한 다음 7단계로 진행합니다.

Router#플랫폼 하드웨어 qfp 활성 데이터 경로 사용률을 표시합니다.

CPU 0: 하위 디바이스 0 5초 1분 5분 60분

입력: 우선 순위(pps) 0 0 0 0

(bps) 208 176 176 176

비우선순위(pps) 0 2 2 2

(bps) 64 784 784 784

합계(pps) 0 2 2 2

(bps) 272 960 960 960

성과: 우선 순위(pps) 0 0 0 0

(bps) 192 160 160 160

비우선순위(pps) 0 1 1 1

(bps) 0 6488 6496 6488

합계(pps) 0 1 1 1

(bps) 192 6648 6656 6648

처리 중: 로드(pct) 0 0 0 0

7단계 - 근본 원인 파악 및 해결 방법 확인

CPU를 과도하게 사용한 것으로 관찰되는 프로세스 또는 프로세스를 식별하면 왜 높은 CPU가 발생했는지에 대한 더 명확한 그림이 나타납니다. 계속 진행하기 위해, 확인된 프로세스에 의해 수행되는 기능을 연구한다. 이렇게 하면 문제에 접근하는 방법에 대한 실행 계획을 결정하는 데 도움이 됩니다. 예 - 프로세스가 특정 프로토콜을 담당하는 경우 이 프로토콜과 관련된 컨피그레이션을 살펴볼 수 있습니다.

그래도 CPU 관련 문제가 발생할 경우 엔지니어가 추가 트러블슈팅을 지원할 수 있도록 TAC에 문의하는 것이 좋습니다. 위의 트러블슈팅 단계는 엔지니어가 문제를 더 효율적으로 격리하는 데 도움이 됩니다.

트러블슈팅 예

이 예에서는 트러블슈팅을 위한 프로세스를 실행하고 라우터의 높은 CPU에 대한 가능한 근본 원인을 가장 잘 파악하려고 합니다. 먼저 어떤 모듈에서 높은 CPU가 사용되는지 파악하기 위해 다음과 같이 출력합니다.

Router#**show platform software status control-processor** 개요

로드 평균

슬롯 상태 1분 5분 15분

RP0 정상 0.66 0.15 0.05

ESP0 정상 0.00 0.00 0.00

SIP0 정상 0.00 0.00 0.00

메모리(kB)

슬롯 상태 총 사용됨(Pct) 사용 가능(Pct) 커밋됨(Pct)

RP0 정상 2009376 1879196 (94%) 130180 (6%) 1432756 (71%)

ESP0 정상 2009400 692472 (34%) 1316928 (66%) 472668 (24%)

SIP0 정상 471804 284556 (60%) 187248 (40%) 193148 (41%)

CPU 사용률

```
슬롯 CPU 사용자 시스템 Nice Idle IRQ SIRQ IOWait
RP0 0 57.11 14.42 0.00 0.00 28.25 0.19 0.00
ESP0 0 2.10 17.91 0.00 79.97 0.00 0.00 0.00
SIP0 0 1.20 6.00 0.00 92.80 0.00 0.00 0.00
```

RP0 내의 유휴 양이 매우 적으므로 Route Processor 내에서 높은 CPU 문제를 나타냅니다. 따라서 추가 트러블슈팅을 위해 RP 내에서 어떤 프로세서가 높은 CPU를 경험하는 것으로 관찰되었는지 파악합니다.

Router#정렬된 프로세스 표시

```
5초 동안의 CPU 사용률: 84%/36%; 1분: 34%; 5분: 9%
PID Runtime(ms) Invoked uSec 5Sec 1Min 5Min TTY Process
107 303230 50749 5975 46.69% 18.12% 4.45% 0 IOSXE-RP Punt Se
63 105617 540091 195 0.23% 0.10% 0.08% 0 IOSD ipc 작업
159 74792 2645991 28 0.15% 0.06% 0.06% 0 VRRS 주 스레드
116 53685 169683 316 0.15% 0.05% 0.01% 0 Per-Second 작업
9 305547 26511 11525 0.15% 0.28% 0.16% 0 체크heap
188 362507 20979154 17 0.15% 0.15% 0.19% 0 Ethernet Msec Ti
3 147 186 790 0.07% 0.08% 0.02% 0 Exec
2 32126 33935 946 0.07% 0.03% 0.00% 0 Load Meter
446 416 33932 12 0.07% 0.00% 0.00% 0VDC 프로세스
164 59945 5261819 11 0.07% 0.04% 0.02% 0 IP ARP 재시도 기간
43 1703 16969 100 0.07% 0.00% 0.00% 0 IPC Keep Alive M
```

이 출력에서 총 CPU 백분율과 인터럽트 백분율이 예상보다 높음을 알 수 있습니다. CPU를 사용하는 최상위 프로세스는 "IOSXE-RP Punt Se"입니다. 이 프로세스는 RP CPU에 대한 트래픽을 처리하는 프로세스이므로 RP에 적용되는 이 트래픽에 대해 자세히 살펴볼 수 있습니다.

라우터#show platform software infrastructure punt

LSMPI 인터페이스 내부 통계:

enabled=0, disabled=0, throttled=0, unthrottled=0, 상태가 준비됨

입력 버퍼 = 90100722

출력 버퍼 = 100439

rxdone 수 = 90100722

txdone 수 = 100436

Rx no particletype count = 0

Tx no particletype count = 0

새도 수의 Txbuf = 0

패킷 시작 없음 = 0

패킷 끝 없음 = 0

Punt 삭제 통계:

잘못된 버전 0

잘못된 유형 0

기능 헤더 0이 있음

플랫폼 헤더 0이 있음
 기능 헤더 누락 0
 공통 헤더 불일치 0
 잘못된 총 길이 0
 잘못된 패킷 길이 0
 잘못된 네트워크 오프셋 0
 Not punt 헤더 0
 알 수 없는 링크 유형 0
 swidb 없음 1
 잘못된 ESS 기능 헤더 0
 ESS 기능 없음 0
 SSLVPN 기능 없음 0
 Punt For Us 유형 알 수 없음 0
 펀트 원인이 0을 벗어남
 IOSXE-RP Punt 패킷의 원인:
 62210226 Layer2 제어 및 레거시 패킷
 147 ARP 요청 또는 응답 패킷
 27801234 데이터 패킷 확인
 84426 RP<->QFP 킥얼라이브 패킷
 인접성 패킷 6개 확보
 1647 For-us 제어 패킷

FOR_US 제어 IPv4 프로토콜 통계:

1647 OSPF 패킷

패킷 히스토그램(500바이트/bin), 평균 크기(92개, 56개):

Pak-Size In-Count 아웃 카운트

0+: 90097805 98790

500명 이상: 0 7

이 출력에서 라우터로 향하는 트래픽을 나타내는 "For-us data packets"에 많은 양의 패킷이 있는 것을 확인할 수 있습니다. 이 카운터는 몇 분 동안 명령 관찰에서 여러 번 증가한 것으로 확인되었습니다. 이렇게 하면 CPU가 종종 컨트롤 플레인 트래픽인 많은 양의 펀트된 트래픽에서 과도하게 사용됩니다. 컨트롤 플레인 트래픽에는 ARP, SSH, SNMP, 경로 업데이트(BGP, EIGRP, OSPF) 등이 포함될 수 있습니다. 이 정보를 통해 높은 CPU의 잠재적 원인을 파악할 수 있으며, 근본 원인을 트러블슈팅하는 데 도움이 됩니다. 예를 들어, 패킷 캡처 또는 다른 트래픽의 모니터를 구현하여 RP에 정확한 트래픽을 보낼 수 있습니다. 그러면 향후 유사한 문제를 방지하기 위해 근본 원인을 파악하고 해결할 수 있습니다.

패킷 캡처가 완료되면 잠재적 Punted 트래픽의 예는 다음과 같습니다.

- ARP: 이는 ARP 요청 수가 너무 많기 때문일 수 있습니다. 이는 여러 IP 주소가 브로드캐스트 인터페이스에 대한 IP 경로 컨피그레이션을 통해 ARP 요청을 전송하는 경우 발생할 수 있습니다. 이는 ARP 테이블의 항목이 플러시되었기 때문일 수도 있으며, 오래된 MAC 주소 항목 또는 작동/작동 중지 인터페이스에 따라 다시 학습해야 합니다.
- SSH: 이 경우 대규모 show 명령(show tech-support)이나 많은 debug 명령이 활성화된 경우 SSH 세션을 통해 많은 CLI가 전송되기 때문에 CPU가 높아질 수 있습니다.

- SNMP: 이는 요청을 처리하는 데 오랜 시간이 걸리므로 CPU가 높아지는 SNMP 에이전트 때문일 수 있습니다. 폴링되는 MIB 또는 NMS에 의해 폴링되는 경로 및/또는 ARP 테이블이 종종 두 가지 원인으로 추정됩니다.
- 경로 업데이트: 네트워크 재통합 또는 링크 플랩 때문에 경로 업데이트가 유발되는 경우가 많습니다. 이는 네트워크 내에서 중단되는 경로 또는 중단되는 전체 디바이스를 나타낼 수 있으며, 이로 인해 네트워크가 통합되고 사용 중인 라우팅 프로토콜에 따라 최상의 경로가 재계산됩니다.

이는 개별 프로세스 수준으로 내려갈 때 높은 CPU의 원인을 파악하여 근본 원인을 찾아내는 방법을 강조합니다. 여기에서 개별 프로세스 또는 프로토콜을 개별적으로 분석하여 컨피그레이션 문제, 소프트웨어 문제, 네트워크 설계 또는 의도된 실무 중 어떤 것인지 파악할 수 있습니다.

추가 명령

다음은 활용할 수 있는 기타 유용한 명령의 목록이며 관련된 프로세서별로 정렬되어 있습니다.

Route Processor

- <프로세스 cpu 기록 표시>
 - 지난 60초, 분, 72시간 동안의 CPU 기록 그래프 제공
- <show process process_ID>
 - 개별 프로세스 메모리 및 CPU 할당에 대한 자세한 정보
- <플랫폼 소프트웨어 인프라 펀트 표시>
 - RP에 적용되는 모든 트래픽에 대한 정보를 제공합니다.
- <show platform software status control-processor brief>
 - CPU의 로드 및 '상태'를 자세히 설명하고 메모리 및 모듈 통계도 자세히 설명합니다
- <show platform software process slot r0|r1 monitor>
 - 선택한 모듈에서 여러 프로세스와 해당 CPU 및 메모리 할당을 자세히 설명합니다
- <모니터링 플랫폼 소프트웨어 프로세스 r0|r1>
 - CPU를 사용할 때 프로세스를 업데이트하는 라이브 피드를 제공합니다.
 - 올바르게 작동하려면 먼저 글로벌 컨피그레이션 모드에서 "terminal terminal-type" 명령을 입력해야 합니다.

Embedded Services Processor

- <show platform software process list fp active summary>
 - CPU에서 실행되는 모든 프로세스의 요약과 평균 로드를 자세히 설명합니다
- <show platform software process slot f0|f1 monitor>
 - 선택한 모듈에서 여러 프로세스와 해당 CPU 및 메모리 할당을 자세히 설명합니다
- <플랫폼 소프트웨어 프로세스 모니터링 f0|f1>
 - 프로세스가 CPU를 사용하는 대로 프로세스를 업데이트하는 라이브 피드를 제공합니다
 - 올바르게 작동하려면 먼저 글로벌 컨피그레이션 모드에서 "terminal terminal-type" 명령을 입력해야 합니다.

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.