

**Oracle® Solaris 11.1에서 시스템 정보,  
프로세스 및 성능 관리**

Copyright © 1998, 2012, Oracle and/or its affiliates. All rights reserved.

본 소프트웨어와 관련 문서는 사용 제한 및 기밀 유지 규정을 포함하는 라이선스 계약서에 의거해 제공되며, 지적 재산법에 의해 보호됩니다. 라이선스 계약서 상에 명시적으로 허용되어 있는 경우나 법규에 의해 허용된 경우를 제외하고, 어떠한 부분도 복사, 재생, 번역, 방송, 수정, 라이선스, 전송, 배포, 진열, 실행, 발행, 또는 전시될 수 없습니다. 본 소프트웨어를 리버스 엔지니어링, 디스어셈블리 또는 디컴파일하는 것은 상호 운용에 대한 법규에 의해 명시된 경우를 제외하고는 금지되어 있습니다.

이 안의 내용은 사전 공지 없이 변경될 수 있으며 오류가 존재하지 않음을 보증하지 않습니다. 만일 오류를 발견하면 서면으로 통지해 주시기 바랍니다.

만일 본 소프트웨어나 관련 문서를 미국 정부나 또는 미국 정부를 대신하여 라이선스한 개인이나 법인에게 배송하는 경우, 다음 공지 사항이 적용됩니다.

#### U.S. GOVERNMENT END USERS:

Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

본 소프트웨어 혹은 하드웨어는 다양한 정보 관리 애플리케이션의 일반적인 사용을 목적으로 개발되었습니다. 본 소프트웨어 혹은 하드웨어는 개인적인 상해를 초래할 수 있는 애플리케이션을 포함한 본질적으로 위험한 애플리케이션에서 사용할 목적으로 개발되거나 그 용도로 사용될 수 없습니다. 만일 본 소프트웨어 혹은 하드웨어를 위험한 애플리케이션에서 사용할 경우, 라이선스 사용자는 해당 애플리케이션의 안전한 사용을 위해 모든 적절한 비상-안전, 백업, 대비 및 기타 조치를 반드시 취해야 합니다. Oracle Corporation과 그 회사는 본 소프트웨어 혹은 하드웨어를 위험한 애플리케이션에서의 사용으로 인해 발생하는 어떠한 손해에 대해서도 책임지지 않습니다.

Oracle과 Java는 Oracle Corporation 및/또는 그 자회사의 등록 상표입니다. 기타의 명칭들은 각 해당 명칭을 소유한 회사의 상표일 수 있습니다.

Intel 및 Intel Xeon은 Intel Corporation의 상표 내지는 등록 상표입니다. SPARC 상표 일체는 라이선스에 의거하여 사용되며 SPARC International, Inc.의 상표 내지는 등록 상표입니다. AMD, Opteron, AMD 로고, 및 AMD Opteron 로고는 Advanced Micro Devices의 상표 내지는 등록 상표입니다. UNIX는 The Open Group의 등록 상표입니다.

본 소프트웨어 혹은 하드웨어와 관련 문서(설명서)는 제 3자로부터 제공되는 콘텐츠, 제품 및 서비스에 접속할 수 있거나 정보를 제공합니다. Oracle Corporation과 그 자회사는 제 3자의 콘텐츠, 제품 및 서비스와 관련하여 어떠한 책임도 지지 않으며 명시적으로 모든 보증에 대해서도 책임을 지지 않습니다. Oracle Corporation과 그 자회사는 제 3자의 콘텐츠, 제품 및 서비스에 접속하거나 사용으로 인해 초래되는 어떠한 손실, 비용 또는 손해에 대해 어떠한 책임도 지지 않습니다.

# 목차

---

머리말 .....	7
<b>1 시스템 정보 관리(작업) .....</b>	<b>11</b>
시스템 정보 표시 및 변경의 새로운 기능 .....	11
hostname 및 domainname 명령의 향상된 기능 .....	11
시스템 정보 표시 .....	12
시스템 정보 표시(작업 맵) .....	12
시스템 정보를 표시하는 데 사용되는 명령 .....	13
▼시스템의 릴리스 정보를 표시하는 방법 .....	13
▼날짜 및 시간 표시 방법 .....	13
▼시스템의 호스트 ID 번호 표시 방법 .....	14
시스템 구조 및 프로세서 유형 표시 .....	14
▼시스템의 제품 이름 표시 방법 .....	15
▼시스템에 설치된 메모리 표시 방법 .....	16
▼장치에 대한 기본 및 사용자 정의된 등록 정보 값을 표시하는 방법 .....	16
▼시스템 진단 정보를 표시하는 방법 .....	20
칩 다중 스레드 기능에 대한 정보 식별 .....	22
▼시스템의 물리적 프로세서 유형 표시 방법 .....	22
▼시스템의 가상 프로세서 유형 표시 방법 .....	23
시스템 정보 변경 .....	24
시스템 정보 변경(작업 맵) .....	24
▼시스템의 날짜 및 시간을 수동으로 설정하는 방법 .....	24
▼오늘의 메시지 설정 방법 .....	25
▼시스템의 ID를 변경하는 방법 .....	25
<b>2 시스템 프로세스 관리(작업) .....</b>	<b>27</b>
시스템 프로세스 관리의 새로운 기능 .....	27

의사 시스템 프로세스 .....	27
시스템 프로세스 관리 .....	28
시스템 프로세스 관리(작업 맵) .....	28
시스템 프로세스 관리 명령 .....	28
프로세스 클래스 정보 관리 .....	37
프로세스 클래스 정보 관리(작업 맵) .....	38
프로세스의 예약 우선 순위 변경(priocntl) .....	39
▼ 프로세스 클래스에 대한 기본 정보를 표시하는 방법(priocntl) .....	39
▼ 프로세스의 전역 우선 순위를 표시하는 방법 .....	39
▼ 프로세스 우선 순위를 지정하는 방법(priocntl) .....	41
▼ 시간 공유 프로세스의 예약 매개변수를 변경하는 방법(priocntl) .....	42
▼ 프로세스의 클래스를 변경하는 방법(priocntl) .....	42
시간 공유 프로세스의 우선 순위 변경(nice) .....	43
▼ 프로세스의 우선 순위를 변경하는 방법(nice) .....	44
시스템 프로세스의 문제 해결 .....	45
<b>3 시스템 성능 모니터링(작업) .....</b>	<b>47</b>
시스템 성능 작업의 위치 .....	47
시스템 성능 및 시스템 리소스 .....	48
프로세스 및 시스템 성능 .....	48
시스템 성능 모니터링 정보 .....	50
모니터링 도구 .....	50
시스템 성능 정보 표시 .....	51
시스템 성능 정보 표시(작업 맵) .....	51
가상 메모리 통계 표시(vmstat) .....	52
디스크 사용률 정보 표시(iostat) .....	55
디스크 공간 통계 표시(df) .....	57
시스템 작업 모니터링 .....	59
시스템 작업 모니터링(작업 맵) .....	59
시스템 작업 모니터링(sar) .....	60
자동으로 시스템 작업 데이터 수집(sar) .....	76
<b>4 시스템 작업 예약(작업) .....</b>	<b>81</b>
시스템 작업을 자동으로 실행하는 방법 .....	81
반복적 작업 예약: crontab .....	82

단일 작업 예약: at .....	82
시스템 작업 예약 .....	83
crontab 파일 만들기 및 편집(작업 맵) .....	83
반복적 시스템 작업 예약(cron) .....	84
crontab 파일 만들기 및 편집 .....	86
crontab 파일 표시 .....	88
crontab 파일 제거 .....	89
crontab 명령에 대한 액세스 제어 .....	90
at 명령을 사용하여 작업 예약 .....	93
at 명령 사용(작업 맵) .....	93
단일 시스템 작업 예약(at) .....	94
<b>5 시스템 콘솔, 터미널 장치 및 전원 서비스 관리(작업) .....</b>	<b>101</b>
시스템 콘솔, 터미널 장치 및 전원 서비스를 관리하기 위한 새로운 기능 .....	101
시스템 전원 서비스 관리 방법의 변경 사항 .....	101
시스템 콘솔 및 로컬로 연결된 터미널 장치 관리 .....	102
시스템 콘솔 및 로컬로 연결된 터미널 장치를 관리하는 SMF 서비스 .....	102
시스템 전원 서비스 관리 .....	105
시스템 전원 문제 해결 .....	107
색인 .....	109



# 머리말

---

**Oracle Solaris 11.1의 시스템 정보, 프로세스 및 성능 관리**는 Oracle Solaris 시스템 관리 정보의 중요한 부분을 다루는 설명서 모음의 일부입니다. 본 설명서에서는 SPARC 기반 시스템과 x86 기반 시스템에 대한 정보를 모두 다룹니다.

본 설명서에서는 다음 작업을 완료한 것으로 가정합니다.

- Oracle Solaris 소프트웨어 설치
- 사용할 모든 네트워킹 소프트웨어 설정

Oracle Solaris의 경우 해당 장의 **새로운 기능**이라는 단원에서 시스템 관리자에게 유용한 새로운 기능에 대해 설명합니다.

---

주 - 본 Oracle Solaris 릴리스는 프로세서 아키텍처의 SPARC 및 x86 제품군을 사용하는 시스템을 지원합니다. 지원되는 시스템은 **Oracle Solaris OS: 하드웨어 호환성 목록**을 참조하십시오. 이 설명서에서는 플랫폼 유형에 따른 구현 차이가 있는 경우 이에 대하여 설명합니다.

이 문서에서 사용되는 x86 관련 용어의 의미는 다음과 같습니다.

- x86은 64비트 및 32비트 x86 호환 제품을 아우르는 큰 제품군을 의미합니다.
- x64는 특히 64비트 x86 호환 CPU와 관련됩니다.
- "32비트 x86"은 x86 기반 시스템에 대한 특정 32비트 정보를 나타냅니다.

지원되는 시스템은 **Oracle Solaris OS: 하드웨어 호환성 목록**을 참조하십시오.

---

## 이 설명서의 대상

본 설명서는 Oracle Solaris 11 릴리스가 실행되고 있는 한 대 이상의 시스템을 관리하는 사용자를 대상으로 합니다. 본 설명서를 사용하려면 1-2년 정도의 UNIX 시스템 관리 경험이 있어야 합니다. UNIX 시스템 관리 교육 과정에 참석하는 것도 도움이 될 수 있습니다.

## Oracle Support에 액세스

Oracle 고객은 My Oracle Support를 통해 온라인 지원에 액세스할 수 있습니다. 자세한 내용은 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>를 참조하거나, 청각 장애가 있는 경우 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>를 방문하십시오.

## 활자체 규약

다음 표는 이 설명서에서 사용되는 활자체 규약에 대해 설명합니다.

표 P-1 활자체 규약

활자체 또는 기호	설명	예제
AaBbCc123	명령, 파일, 디렉토리 이름 및 컴퓨터 화면에 출력되는 내용입니다.	.login 파일을 편집하십시오. 모든 파일 목록을 보려면 <code>ls -a</code> 명령을 사용하십시오. <code>machine_name% you have mail.</code>
AaBbCc123	사용자가 입력하는 내용으로 컴퓨터 화면의 출력 내용과 대조됩니다.	<code>machine_name% su</code> Password:
AaBbCc123	위치 표시자: 실제 이름이나 값으로 바뀝니다.	<code>rm filename</code> 명령을 사용하여 파일을 제거합니다.
AaBbCc123	설명서 제목, 새 용어, 강조 표시할 용어입니다.	<b>사용자 설명서</b> 의 6장을 읽으십시오. <b>캐시</b> 는 로컬로 저장된 복사본입니다. 파일을 저장하면 <b>안 됩니다</b> . <b>주:</b> 일부 강조된 항목은 온라인에서 굵은체로 나타납니다.

## 명령 예의 셸 프롬프트

다음 표에는 Oracle Solaris OS에 포함된 셸의 기본 UNIX 시스템 프롬프트 및 슈퍼유저 프롬프트가 나와 있습니다. 명령 예제에 표시된 기본 시스템 프롬프트는 Oracle Solaris 릴리스에 따라 다릅니다.

표 P-2 셸 프롬프트

셸	프롬프트
Bash 셸, Korn 셸 및 Bourne 셸	\$
수퍼유저용 Bash 셸, Korn 셸 및 Bourne 셸	#
C 셸	machine_name%
수퍼유저용 C 셸	machine_name#



## 시스템 정보 관리(작업)

---

이 장에서는 가장 일반적인 시스템 정보를 표시하고 변경하는 데 필요한 작업에 대해 설명합니다.

이 장에서는 시스템 리소스를 유연하게 할당, 모니터링 및 제어하는 리소스 관리에 대한 정보는 다루지 않습니다. 리소스 관리를 사용하여 시스템 리소스를 관리하는 방법에 대한 자세한 내용은 **Oracle Solaris 11.1 관리: Oracle Solaris 영역, Oracle Solaris 10 영역 및 리소스 관리의 1 장**, “리소스 관리 소개”를 참조하십시오.

다음은 이 장에 포함된 정보 목록입니다.

- 11 페이지 “시스템 정보 표시 및 변경의 새로운 기능”
- 12 페이지 “시스템 정보 표시”
- 24 페이지 “시스템 정보 변경”

### 시스템 정보 표시 및 변경의 새로운 기능

#### hostname 및 domainname 명령의 향상된 기능

이 릴리스에서 `hostname` 및 `domainname` 명령은 `hostname` 및 `domainname`을 보다 쉽게 영구적으로 설정할 수 있도록 향상되었습니다. 이러한 명령을 사용하면 해당 SMF 등록 정보 및 연관된 SMF 서비스도 자동으로 업데이트됩니다.

자세한 내용은 25 페이지 “시스템의 ID를 변경하는 방법”과 `hostname(1)`, `domainname(1M)` 및 `nodename(4)` 매뉴얼 페이지를 참조하십시오.

# 시스템 정보 표시

다음 표는 일반 시스템 정보를 표시하는 데 사용할 수 있는 명령을 설명합니다.

## 시스템 정보 표시(작업 맵)

작업	설명	수행 방법
시스템의 릴리스 정보를 표시합니다.	/etc/release 파일의 콘텐츠를 표시하여 Oracle Solaris 릴리스 버전을 식별합니다.	13 페이지 “시스템의 릴리스 정보를 표시하는 방법”
시스템의 날짜 및 시간을 표시합니다.	date 명령을 사용하여 시스템의 날짜 및 시간을 표시합니다.	13 페이지 “날짜 및 시간 표시 방법”
시스템의 호스트 ID 번호를 표시합니다.	hostid 명령을 사용하여 시스템의 호스트 ID를 표시합니다.	14 페이지 “시스템의 호스트 ID 번호 표시 방법”
시스템의 구조 또는 프로세서 유형을 표시합니다.	시스템의 구조 유형을 표시하려면 isainfo 명령을 사용합니다. 시스템의 프로세서 유형을 표시하려면 isalist 명령을 사용합니다.	14 페이지 “시스템 구조 및 프로세서 유형 표시”
시스템의 제품 이름을 표시합니다.	prtconf -b 명령을 사용하여 시스템의 제품 이름을 표시할 수 있습니다.	15 페이지 “시스템의 제품 이름 표시 방법”
시스템에 설치된 메모리를 표시합니다.	prtconf 명령을 사용하여 시스템에 설치된 메모리에 대한 정보를 표시합니다.	16 페이지 “시스템에 설치된 메모리 표시 방법”
장치에 대한 원본 및 기본 등록 정보 값을 표시합니다.	prtconf 명령을 -u 옵션과 함께 사용하여 장치에 대한 기본 및 업데이트된 등록 정보 값을 모두 표시합니다.	16 페이지 “장치에 대한 기본 및 사용자 정의된 등록 정보 값을 표시하는 방법”
시스템의 구성 및 진단 정보를 표시합니다.	시스템 구성 및 진단 정보를 표시하려면 적합한 옵션과 함께 prtdiag 명령을 사용합니다.	20 페이지 “시스템 진단 정보를 표시하는 방법”

작업	설명	수행 방법
시스템의 물리 및 가상 프로세서 정보를 표시합니다.	<p><code>psrinfo -p</code> 명령을 사용하여 시스템의 총 물리적 프로세서 수를 나열합니다.</p> <p><code>psrinfo -pv</code> 명령을 사용하여 시스템의 모든 물리적 프로세서와 각 물리적 프로세서에 연결된 가상 프로세서를 나열합니다.</p>	22 페이지 “시스템의 물리적 프로세서 유형 표시 방법”

## 시스템 정보를 표시하는 데 사용되는 명령

표 1-1 시스템 정보 표시 명령

명령	표시되는 시스템 정보	매뉴얼 페이지
<code>date</code>	날짜와 시간	<code>date(1)</code>
<code>hostid</code>	호스트 ID 번호	<code>hostid(1)</code>
<code>isainfo</code>	실행 중인 시스템에서 <b>고유</b> 응용 프로그램이 지원하며 스크립트에 토큰으로 전달될 수 있는 비트 수	<code>isainfo(1)</code>
<code>isalist</code>	프로세서 유형	<code>isalist(1)</code>
<code>prtconf</code>	시스템 구성 정보, 설치된 메모리, 장치 등록 정보 및 제품 이름	<code>prtconf(1M)</code>
<code>prtdiag</code>	결함이 발생한 FRU(현장 대체 가능 장치)를 포함한 시스템 구성 및 진단 정보	<code>prtdiag(1M)</code>
<code>psrinfo</code>	프로세서 정보	<code>psrinfo(1M)</code>
<code>uname</code>	운영 체제 이름, 릴리스, 버전, 노드 이름, 하드웨어 이름 및 프로세서 유형	<code>uname(1)</code>

### ▼ 시스템의 릴리스 정보를 표시하는 방법

- `/etc/release` 파일의 콘텐츠를 표시하여 릴리스 버전을 식별합니다.  
\$ `cat /etc/release`

### ▼ 날짜 및 시간 표시 방법

- 시스템 클럭에 따라 현재 날짜 및 시간을 표시하려면 `date` 명령을 사용합니다.

### 예 1-1 날짜 및 시간 표시

다음 예에서는 `date` 명령의 샘플 출력을 보여 줍니다.

```
$ date
Fri Jun 1 16:07:44 MDT 2012
$
```

## ▼ 시스템의 호스트 ID 번호 표시 방법

- 호스트 ID 번호를 숫자(16진수) 형식으로 표시하려면 `hostid` 명령을 사용합니다.

### 예 1-2 시스템의 호스트 ID 번호 표시

다음 예에서는 `hostid` 명령의 샘플 출력을 보여 줍니다.

```
$ hostid
80a5d34c
```

## 시스템 구조 및 프로세서 유형 표시

다음 예제에서는 x86 및 SPARC 기반 시스템에서 `isainfo` 및 `isalist` 명령을 실행할 때의 결과를 보여 줍니다.

### 예 1-3 시스템의 구조 유형 표시

다음 예제에서는 현재 운영 체제에서 지원되는 응용 프로그램에 대한 구조 유형 및 고유 명령 세트를 표시하는 방법을 보여 줍니다.

다음은 x86 기반 시스템의 출력입니다.

```
$ isainfo
amd64 i386
```

다음은 SPARC 기반 시스템의 출력입니다.

```
$ isainfo
sparcv9 sparc
```

`isainfo -v` 명령은 32비트 및 64비트 응용 프로그램 지원을 표시합니다. 예를 들어, SPARC 기반 시스템의 출력은 다음과 같습니다.

```
$ isainfo -v
64-bit sparcv9 applications
    asi_blk_init
32-bit sparc applications
    asi_blk_init v8plus div32 mul32
#
```

### 예 1-3 시스템의 구조 유형 표시 (계속)

이 예제에서는 x86 기반 시스템의 `isainfo -v` 명령 출력을 보여줍니다.

```
$ isainfo -v
64-bit amd64 applications
    sse4.1 ssse3 ahf cx16 sse3 sse2 sse fxsr mmx cmov amd_sysc cx8 tsc fpu
32-bit i386 applications
    sse4.1 ssse3 ahf cx16 sse3 sse2 sse fxsr mmx cmov sep cx8 tsc fpu
```

`isainfo(1)` 매뉴얼 페이지를 참조하십시오.

### 예 1-4 시스템의 프로세서 유형 표시

다음 예에서는 x86 기반 시스템의 프로세서 유형에 대한 정보를 표시하는 방법을 보여줍니다.

```
$ isalist
pentium_pro+mmx pentium_pro pentium+mmx pentium i486 i386 i86
```

다음 예에서는 SPARC 기반 시스템의 프로세서 유형에 대한 정보를 표시하는 방법을 보여줍니다.

```
$ isalist
sparcv9 sparcv8plus sparcv8 sparcv8-fsmuld sparcv7 sparc sparcv9+vis sparcv9+vis2 \
sparcv8plus+vis sparcv8plus+vis2
```

`isalist(1)` 매뉴얼 페이지를 참조하십시오.

## ▼ 시스템의 제품 이름 표시 방법

`prtconf` 명령의 `-b` 옵션을 사용하면 시스템의 제품 이름을 표시할 수 있습니다. 자세한 내용은 `prtconf(1M)` 매뉴얼 페이지를 참조하십시오.

- 시스템의 제품 이름을 표시하려면 다음과 같이 `-b` 옵션과 함께 `prtconf` 명령을 사용합니다.

```
$ prtconf -b
```

### 예 1-5 SPARC: 시스템의 제품 이름 표시

다음 예에서는 SPARC 기반 시스템의 `prtconf -b` 명령의 샘플 출력을 보여 줍니다.

```
$ prtconf -b
name: ORCL,SPARC-T4-2
banner-name: SPARC T4-2
compatible: 'sun4v'
$
```

다음 예에서는 SPARC 기반 시스템의 `prtconf -vb` 명령의 샘플 출력을 보여 줍니다.

```
$ prtconf -vb
name: ORCL,SPARC-T3-4
banner-name: SPARC T3-4
compatible: 'sun4v'
idprom: 01840014.4fa02d28.00000000.a02d28de.00000000.00000000.00000000.00000000
openprom model: SUNW,4.33.0.b
openprom version: 'OBP 4.33.0.b 2011/05/16 16:26'
```

## ▼ 시스템에 설치된 메모리 표시 방법

- 시스템에 설치된 메모리를 표시하려면 `prtconf` 명령을 사용합니다.

### 예 1-6 시스템에 설치된 메모리 표시

다음 예에서는 `prtconf` 명령의 샘플 출력을 보여줍니다. `grep Memory` 명령은 메모리 정보만 표시하기 위해 `prtconf` 명령의 출력을 선택합니다.

```
$ prtconf | grep Memory
Memory size: 523776 Megabytes
```

## ▼ 장치에 대한 기본 및 사용자 정의된 등록 정보 값을 표시하는 방법

장치에 대한 기본 및 사용자 정의된 등록 정보 값을 모두 표시하려면 `prtconf` 명령을 `-u` 옵션과 함께 사용합니다. 이 옵션에 대한 자세한 내용은 `prtconf(1M)` 매뉴얼 페이지를 참조하십시오.

- `driver.conf` 파일의 기본 및 사용자 정의된 등록 정보를 표시합니다.

```
$ prtconf -u
```

`prtconf -u` 명령 출력에는 시스템에 있는 모든 드라이버에 대한 기본 및 사용자 정의된 등록 정보가 표시됩니다.

### 예 1-7 SPARC: 기본 및 사용자 정의 장치 등록 정보 표시

이 예는 `bge.conf` 파일에 대한 기본 및 사용자 정의 등록 정보를 보여줍니다. 공급업체에서 제공하는 구성 파일은 `/kernel` 및 `/platform` 디렉토리에 있고 수정된 해당 드라이버 구성 파일은 `/etc/driver/drv` 디렉토리에 있습니다.

```
$ prtconf -u
System Configuration: Oracle Corporation sun4v
Memory size: 523776 Megabytes
System Peripherals (Software Nodes):

ORCL,SPARC-T3-4
```

```

scsi_vhci, instance #0
  disk, instance #4
  disk, instance #5
  disk, instance #6
  disk, instance #8
  disk, instance #9
  disk, instance #10
  disk, instance #11
  disk, instance #12
packages (driver not attached)
  SUNW,builtin-drivers (driver not attached)
  deblocker (driver not attached)
  disk-label (driver not attached)
  terminal-emulator (driver not attached)
  dropins (driver not attached)
  SUNW,asr (driver not attached)
  kbd-translator (driver not attached)
  obp-tftp (driver not attached)
  zfs-file-system (driver not attached)
  hsfs-file-system (driver not attached)
chosen (driver not attached)
openprom (driver not attached)
  client-services (driver not attached)
options, instance #0
aliases (driver not attached)
memory (driver not attached)
virtual-memory (driver not attached)
iscsi-hba (driver not attached)
  disk, instance #0 (driver not attached)
virtual-devices, instance #0
  flashprom (driver not attached)
  tpm, instance #0 (driver not attached)
  n2cp, instance #0
  ncp, instance #0
  random-number-generator, instance #0
  console, instance #0
  channel-devices, instance #0
    virtual-channel, instance #0
    virtual-channel, instance #1
    virtual-channel-client, instance #2
    virtual-channel-client, instance #3
    virtual-domain-service, instance #0
cpu (driver not attached)

```

### 예 1-8 x86: 기본 및 사용자 정의 장치 등록 정보 표시

이 예는 `bge.conf` 파일에 대한 기본 및 사용자 정의 등록 정보를 보여줍니다. 공급업체에서 제공하는 구성 파일은 `/kernel` 및 `/platform` 디렉토리에 있고 수정된 해당 드라이버 구성 파일은 `/etc/driver/drv` 디렉토리에 있습니다.

```
$ prtconf -u
System Configuration: Oracle Corporation i86pc
Memory size: 8192 Megabytes
System Peripherals (Software Nodes):
```

```
i86pc
  scsi_vhci, instance #0
  pci, instance #0
    pci10de,5e (driver not attached)
  isa, instance #0
    asy, instance #0
    motherboard (driver not attached)
    pit_bEEP, instance #0
  pci10de,cb84 (driver not attached)
  pci108e,cb84, instance #0
    device, instance #0
      keyboard, instance #0
      mouse, instance #1
  pci108e,cb84, instance #0
  pci-ide, instance #0
    ide, instance #0
      sd, instance #0
    ide (driver not attached)
  pci10de,5c, instance #0
    display, instance #0
  pci10de,cb84, instance #0
  pci10de,5d (driver not attached)
  pci10de,5d (driver not attached)
  pci10de,5d (driver not attached)
  pci10de,5d (driver not attached)
  pci1022,1100, instance #0
  pci1022,1101, instance #1
  pci1022,1102, instance #2
  pci1022,1103 (driver not attached)
  pci1022,1100, instance #3
  pci1022,1101, instance #4
  pci1022,1102, instance #5
  pci1022,1103 (driver not attached)
  pci, instance #1
    pci10de,5e (driver not attached)
    pci10de,cb84 (driver not attached)
    pci10de,cb84, instance #1
    pci10de,5d (driver not attached)
    pci10de,5d (driver not attached)
    pci10de,5d (driver not attached)
    pci10de,5d (driver not attached)
    pci1022,7458, instance #1
    pci1022,7459 (driver not attached)
    pci1022,7458, instance #2
      pci8086,1011, instance #0
      pci8086,1011, instance #1
      pci1000,3060, instance #0
        sd, instance #1
        sd, instance #2
    pci1022,7459 (driver not attached)
  ioapics (driver not attached)
  ioapic, instance #0 (driver not attached)
  ioapic, instance #1 (driver not attached)
  fw, instance #0
```

```

cpu (driver not attached)
cpu (driver not attached)
cpu (driver not attached)
cpu (driver not attached)
sb, instance #1
used-resources (driver not attached)
iscsi, instance #0
fcoe, instance #0
pseudo, instance #0
options, instance #0
xsvc, instance #0
vga_arbiter, instance #0

```

### 예 1-9 x86: 시스템 구성 정보 표시

다음 예에서는 x86 기반 시스템에서 `prtconf` 명령과 `-v` 옵션을 사용하여 시스템에 연결된 디스크, 테이프 및 DVD 장치를 식별하는 방법을 보여줍니다. 이 명령의 출력은 장치 인스턴스 옆에 드라이버가 연결되지 않음 메시지를 표시합니다. 이 메시지는 일반적으로 해당 장치 인스턴스에 장치가 없음을 의미합니다.

```

$ prtconf -v | more
System Configuration: Oracle Corporation i86pc
Memory size: 8192 Megabytes
System Peripherals (Software Nodes):

i86pc
  System properties:
    name='#size-cells' type=int items=1
      value=00000002
    name='#address-cells' type=int items=1
      value=00000003
    name='relative-addressing' type=int items=1
      value=00000001
    name='MMU_PAGEOFFSET' type=int items=1
      value=00000fff
    name='MMU_PAGESIZE' type=int items=1
      value=00001000
    name='PAGESIZE' type=int items=1
      value=00001000
    name='acpi-status' type=int items=1
      value=00000013
    name='biosdev-0x81' type=byte items=588
      value=01.38.74.0e.08.1e.db.e4.fe.00.d0.ed.fe.f8.6b.04.08.d3.db.e4.fe
    .
    .
    .

```

**참조** 자세한 내용은 `driver(4)`, `driver.conf(4)` 및 `prtconf(1M)` 매뉴얼 페이지를 참조하십시오.

관리용으로 추가된 구성 파일을 만드는 방법에 대한 자세한 내용은 [Oracle Solaris 11.1 관리: 장치 및 파일 시스템의 3 장](#), “장치 관리(작업)”를 참조하십시오.

## ▼ 시스템 진단 정보를 표시하는 방법

- 시스템에 대한 구성 및 진단 정보를 표시합니다.

```
$ prtdiag [-v] [-l]
```

-v Verbose 모드입니다.

-l 로그 출력입니다. 결함 또는 오류가 시스템에 존재하는 경우 이 정보를 syslogd(1M)에만 출력합니다.

### 예 1-10 SPARC: 시스템 진단 정보 표시

다음 예에서는 SPARC 기반 시스템의 `prtdiag -v` 명령에 대한 출력을 보여줍니다. 이 예제는 간략하게 보이기 위해 내용이 잘려있습니다.

```
$ prtdiag -v | more
```

```
System Configuration: Oracle Corporation sun4v Sun Fire T200
Memory size: 16256 Megabytes
```

```
===== Virtual CPUs =====
```

CPU ID	Frequency	Implementation	Status
0	1200 MHz	SUNW,UltraSPARC-T1	on-line
1	1200 MHz	SUNW,UltraSPARC-T1	on-line
2	1200 MHz	SUNW,UltraSPARC-T1	on-line
3	1200 MHz	SUNW,UltraSPARC-T1	on-line
4	1200 MHz	SUNW,UltraSPARC-T1	on-line
5	1200 MHz	SUNW,UltraSPARC-T1	on-line
6	1200 MHz	SUNW,UltraSPARC-T1	on-line

.  
.
  
.

```
===== Physical Memory Configuration =====
```

```
Segment Table:
```

Base Address	Segment Size	Interleave Factor	Bank Size	Contains Modules
0x0	16 GB	4	2 GB	MB/CMP0/CH0/R0/D0 MB/CMP0/CH0/R0/D1
			2 GB	MB/CMP0/CH0/R1/D0 MB/CMP0/CH0/R1/D1
			2 GB	MB/CMP0/CH1/R0/D0 MB/CMP0/CH1/R0/D1
			2 GB	MB/CMP0/CH1/R1/D0

.  
.

```
System PROM revisions:
```

```
-----
OBP 4.30.4.d 2011/07/06 14:29
```

```

IO ASIC revisions:
-----
Location          Path          Device
          Revision
-----
IOBD/IO-BRIDGE          /pci@780      SUNW,sun4v-pci      0
.
.
.
    
```

**예 1-11 x86: 시스템 진단 정보 표시**

다음 예에서는 x86 기반 시스템의 `prtdiag -l` 명령에 대한 출력을 보여줍니다.

```

$ prtdiag -l
System Configuration: ... Sun Fire X4100 M2
BIOS Configuration: American Megatrends Inc. 0ABJX104 04/09/2009
BMC Configuration: IPMI 1.5 (KCS: Keyboard Controller Style)

==== Processor Sockets =====
Version          Location Tag
-----
Dual-Core AMD Opteron(tm) Processor 2220 CPU 1
Dual-Core AMD Opteron(tm) Processor 2220 CPU 2

==== Memory Device Sockets =====
Type      Status Set Device Locator      Bank Locator
-----
unknown   empty  0   DIMM0          NODE0
unknown   empty  0   DIMM1          NODE0
DDR2      in use 0   DIMM2          NODE0
DDR2      in use 0   DIMM3          NODE0
unknown   empty  0   DIMM0          NODE1
unknown   empty  0   DIMM1          NODE1
DDR2      in use 0   DIMM2          NODE1
DDR2      in use 0   DIMM3          NODE1

==== On-Board Devices =====
LSI serial-SCSI #1
Gigabit Ethernet #1
ATI Rage XL VGA

==== Upgradeable Slots =====
ID  Status  Type          Description
-----
1   available PCI Express  PCIExp SLOT0
2   available PCI Express  PCIExp SLOT1
3   available PCI-X        PCIX SLOT2
4   available PCI Express  PCIExp SLOT3
5   available PCI Express  PCIExp SLOT4
$
    
```

## 칩 다중 스레드 기능에 대한 정보 식별

psrinfo 명령이 가상 프로세서에 대한 정보에 더하여 실제 프로세서에 대한 정보 또한 제공하도록 수정되었습니다. 이 개선된 기능은 칩 다중 스레드(CMT) 기능을 식별하기 위하여 추가되었습니다. 새 -p 옵션을 사용하면 시스템에 있는 실제 프로세서의 총 수가 보고됩니다. psrinfo -pv 명령을 사용하면 시스템에 있는 실제 프로세서뿐 아니라 각 실제 프로세서에 연결된 가상 프로세서가 모두 목록으로 표시됩니다. psrinfo 명령의 기본 출력에는 시스템의 가상 프로세서 정보가 계속 표시됩니다.

자세한 내용은 psrinfo(1M) 매뉴얼 페이지를 참조하십시오.

이 기능과 연관된 절차에 대한 내용은 22 페이지 “시스템의 물리적 프로세서 유형 표시 방법”을 참조하십시오.

### ▼ 시스템의 물리적 프로세서 유형 표시 방법

- psrinfo -p 명령을 사용하여 시스템의 총 물리적 프로세서 수를 표시합니다.

```
$ psrinfo -p
1
```

psrinfo -pv 명령을 사용하여 시스템의 각 물리적 프로세서 및 각 물리적 프로세서에 연결된 가상 프로세서에 대한 정보를 표시합니다. 예를 들면 다음과 같습니다.

```
$ psrinfo -pv
The physical processor has 8 cores and 32 virtual processors (0-31)
  The core has 4 virtual processors (0-3)
  The core has 4 virtual processors (4-7)
  The core has 4 virtual processors (8-11)
  The core has 4 virtual processors (12-15)
  The core has 4 virtual processors (16-19)
  The core has 4 virtual processors (20-23)
  The core has 4 virtual processors (24-27)
  The core has 4 virtual processors (28-31)
  UltraSPARC-T1 (chipid 0, clock 1000 MHz)
```

x86 기반 시스템에서 psrinfo -pv 명령을 사용할 경우 다음 출력이 표시됩니다.

```
$ psrinfo -pv
The physical processor has 2 virtual processors (0 1)
  x86 (AuthenticAMD 40F13 family 15 model 65 step 3 clock 2793 MHz)
    Dual-Core AMD Opteron(tm) Processor 2220      [ Socket: F(1207) ]
The physical processor has 2 virtual processors (2 3)
  x86 (AuthenticAMD 40F13 family 15 model 65 step 3 clock 2793 MHz)
    Dual-Core AMD Opteron(tm) Processor 2220      [ Socket: F(1207) ]
```

## ▼ 시스템의 가상 프로세서 유형 표시 방법

- `psrinfo -v` 명령을 사용하여 시스템의 가상 프로세서 유형에 대한 정보를 표시합니다.

```
$ psrinfo -v
```

x86 기반 시스템에서 `isalist` 명령을 사용하여 가상 프로세서 유형을 표시합니다. 예를 들면 다음과 같습니다.

```
$ isalist
```

```
amd64 pentium_pro+mmx pentium_pro pentium+mmx pentium i486 i386 i86
```

### 예 1-12 SPARC: 시스템의 가상 프로세서 유형 표시

다음 예에서는 SPARC 기반 시스템의 가상 프로세서 유형에 대한 정보를 표시하는 방법을 보여줍니다.

```
$ psrinfo -v
```

```
Status of virtual processor 28 as of: 09/13/2010 14:07:47
```

```
on-line since 04/08/2010 21:27:56.
```

```
The sparcv9 processor operates at 1400 MHz,
```

```
and has a sparcv9 floating point processor.
```

```
Status of virtual processor 29 as of: 09/13/2010 14:07:47
```

```
on-line since 04/08/2010 21:27:56.
```

```
The sparcv9 processor operates at 1400 MHz,
```

```
and has a sparcv9 floating point processor.
```

### 예 1-13 SPARC: 시스템의 각 물리적 프로세서와 연관된 가상 프로세서 표시

다음 예에서는 Oracle SPARC T4-4 서버에서 `-pv`로 실행할 때의 `psrinfo` 명령 출력을 보여줍니다. 출력에는 칩(물리적 프로세서) 및 스레드 위치에 대한 코어 정보가 모두 표시됩니다. 이 정보는 스레드가 있는 물리적 CPU를 확인하고 코어 레벨에서 어떻게 매핑되었는지를 확인하는 데 유용할 수 있습니다.

```
$ psrinfo -pv
```

```
The physical processor has 8 cores and 64 virtual processors (0-63)
```

```
The core has 8 virtual processors (0-7)
```

```
The core has 8 virtual processors (8-15)
```

```
The core has 8 virtual processors (16-23)
```

```
The core has 8 virtual processors (24-31)
```

```
The core has 8 virtual processors (32-39)
```

```
The core has 8 virtual processors (40-47)
```

```
The core has 8 virtual processors (48-55)
```

```
The core has 8 virtual processors (56-63)
```

```
SPARC-T4 (chipid 0, clock 2998 MHz)
```

```
The physical processor has 8 cores and 64 virtual processors (64-127)
```

```
The core has 8 virtual processors (64-71)
```

```
The core has 8 virtual processors (72-79)
```

```
The core has 8 virtual processors (80-87)
```

```
The core has 8 virtual processors (88-95)
```

```
The core has 8 virtual processors (96-103)
```

```
The core has 8 virtual processors (104-111)
The core has 8 virtual processors (112-119)
The core has 8 virtual processors (120-127)
SPARC-T4 (chipid 1, clock 2998 MHz)
```

## 시스템 정보 변경

이 단원에서는 일반적인 시스템 정보 변경에 사용할 수 있는 명령에 대해 설명합니다.

### 시스템 정보 변경(작업 맵)

작업	지침	수행 방법
시스템의 날짜 및 시간을 수동으로 설정합니다.	<code>date mmddHHMM[[cc]yy]</code> 명령줄 구문을 사용하여 시스템의 날짜 및 시간을 수동으로 설정합니다.	24 페이지 “시스템의 날짜 및 시간을 수동으로 설정하는 방법”
오늘의 메시지를 설정합니다.	<code>/etc/motd</code> 파일을 편집하여 시스템에서 오늘의 메시지를 설정합니다.	25 페이지 “오늘의 메시지 설정 방법”
시스템의 ID를 변경합니다.	<code>hostname</code> 명령을 사용하여 시스템의 ID를 변경합니다.	25 페이지 “시스템의 ID를 변경하는 방법”

## ▼ 시스템의 날짜 및 시간을 수동으로 설정하는 방법

### 1 관리자로 전환합니다.

**Oracle Solaris 11.1 관리: 보안 서비스의 “지정된 관리 권한을 사용하는 방법”**을 참조하십시오.

### 2 새 날짜 및 시간을 입력합니다.

```
$ date mmddHHMM[[cc]yy]
mm    두 자리의 월입니다.
dd    두 자리의 날짜입니다.
HH    두 자리의 24시간 형식 시간입니다.
MM    두 자리의 분입니다.
cc    두 자리의 세기입니다.
```

yy 두 자리의 연도입니다.

자세한 내용은 [date\(1\)](#) 매뉴얼 페이지를 참조하십시오.

- 3 옵션 없이 `date` 명령을 사용하여 시스템의 날짜를 제대로 재설정했는지 확인합니다.

#### 예 1-14 시스템의 날짜 및 시간을 수동으로 설정

다음 예에서는 `date` 명령을 사용하여 시스템의 날짜 및 시간을 수동으로 설정하는 방법을 보여 줍니다.

```
# date
Monday, September 13, 2010 02:00:16 PM MDT
# date 0921173404
Thu Sep 17:34:34 MST 2010
```

## ▼ 오늘의 메시지 설정 방법

사용자가 로그인할 때 시스템의 모든 사용자에게 표시될 공지 또는 문의가 포함되도록 오늘의 메시지 파일인 `/etc/motd`를 편집할 수 있습니다. 이 기능을 자주 사용하지 않을 경우 해당 파일을 정기적으로 편집하여 오래된 메시지를 제거합니다.

- 1 `root` 역할 또는 `Administrator Message Edit` 프로파일이 지정된 역할을 말합니다.

[Oracle Solaris 11.1 관리: 보안 서비스의 “지정된 관리 권한을 사용하는 방법”](#)을 참조하십시오.

- 2 `pfedit` 명령을 사용하여 `/etc/motd` 파일을 편집하고 선택한 메시지를 추가합니다.

```
$ pfedit /etc/motd
```

사용자 로그인 시 표시될 메시지가 포함되도록 텍스트를 편집합니다. 공백, 탭 및 캐리지 리턴을 포함합니다.

- 3 `/etc/motd` 파일의 콘텐츠를 표시하여 변경 사항을 확인합니다.

```
$ cat /etc/motd
```

```
Welcome to the UNIX Universe. Have a nice day.
```

## ▼ 시스템의 ID를 변경하는 방법

- 1 `root` 역할을 말합니다.

[Oracle Solaris 11.1 관리: 보안 서비스의 “지정된 관리 권한을 사용하는 방법”](#)을 참조하십시오.

**2 시스템에 대한 이름 또는 호스트를 설정합니다.**

```
# hostname mynodename
```

## 시스템 프로세스 관리(작업)

---

이 장에서는 시스템 프로세스 관리 절차를 설명합니다.

다음은 이 장에 포함된 정보 목록입니다.

- 27 페이지 “시스템 프로세스 관리의 새로운 기능”
- 28 페이지 “시스템 프로세스 관리”
- 37 페이지 “프로세스 클래스 정보 관리”
- 45 페이지 “시스템 프로세스의 문제 해결”

### 시스템 프로세스 관리의 새로운 기능

이번 릴리스에서 새롭게 제공되었거나 변경된 시스템 프로세스 관리 기능은 다음과 같습니다.

#### 의사 시스템 프로세스

Oracle Solaris 10 및 Oracle Solaris 11 릴리스에는 특정 작업을 수행하지만 어떠한 관리도 필요하지 않은 시스템 프로세스가 포함됩니다.

프로세스	설명
fsflush	페이지를 디스크에 비우는 시스템 데몬입니다.
init	다른 프로세스 및 SMF 구성 요소를 시작 및 재시작하는 초기 시스템 프로세스입니다.
intrad	인터럽트로 인한 시스템 로드를 모니터하고 균형을 조정하는 시스템 프로세스입니다.
kmem_task	메모리 캐시 크기를 모니터하는 시스템 프로세스입니다.

프로세스	설명
pageout	디스크에 대한 메모리 페이징을 제어하는 시스템 프로세스입니다.
sched	OS 예약 및 프로세스 교체를 수행하는 시스템 프로세스입니다.
vm_tasks	성능 향상을 위해 CPU 간 가상 메모리 관련 작업 로드를 분배하고 균형을 조정하는 프로세서당 하나의 스레드가 포함된 시스템 프로세스입니다.
zpool-pool-name	연관된 풀에 대한 I/O 작업 스레드를 포함하는 각 ZFS 저장소 풀에 대한 시스템 프로세스입니다.

## 시스템 프로세스 관리

이 절에서는 시스템 프로세스를 관리하기 위한 여러 작업들에 대해 설명합니다.

### 시스템 프로세스 관리(작업 맵)

작업	설명	수행 방법
프로세스를 나열합니다.	ps 명령을 사용하여 시스템의 모든 프로세스를 나열합니다.	32 페이지 “프로세스를 나열하는 방법”
프로세스에 대한 정보를 표시합니다.	pgrep 명령을 사용하여 추가 정보를 표시하고자 하는 프로세스에 대한 프로세스 ID를 가져옵니다.	33 페이지 “프로세스에 대한 정보를 표시하는 방법”
프로세스를 제어합니다.	pgrep 명령을 사용하여 프로세스를 찾습니다. 그런 다음 적당한 pcommand(/proc)를 사용하여 프로세스를 제어합니다. (표 2-3) 명령에 대한 설명은 Table 2-3을 참조하십시오.	34 페이지 “프로세스를 제어하는 방법”
프로세스를 종료합니다.	프로세스 이름이나 프로세스 ID로 프로세스를 찾습니다. pkill 또는 kill 명령을 사용하여 프로세스를 종료할 수 있습니다.	35 페이지 “프로세스를 종료하는 방법(pkil)” 36 페이지 “프로세스를 종료하는 방법(kill)”

### 시스템 프로세스 관리 명령

다음 표는 시스템 프로세스 관리 명령을 설명합니다.

표 2-1 프로세스 관리 명령

명령	설명	매뉴얼 페이지
ps, pgrep, prstat, pkill	시스템에서 활성 프로세스의 상태를 확인하고, 프로세스에 대한 자세한 정보를 표시합니다.	ps(1), pgrep(1) 및 prstat(1M)
pkill	pgrep와 동일한 기능을 수행하지만, 이름이나 기타 속성으로 프로세스를 찾거나 신호하고 프로세스를 종료합니다. 일치하는 각 프로세스는 해당 프로세스 ID가 출력되는 대신 kill 명령인 것처럼 신호됩니다.	pgrep(1) 및 pkill(1) kill(1)
pargs, preap	프로세스 디버깅을 돕습니다.	pargs(1) 및 preap(1)
dispadm	기본 프로세스 예약 정책을 나열합니다.	dispadm(1M)
priocntl	우선 순위 클래스에 프로세스를 지정하고 프로세스 우선 순위를 관리합니다.	priocntl(1)
nice	시간 공유 프로세스의 우선 순위를 변경합니다.	nice(1)
psrset	특정 프로세스 그룹을 단일 프로세서가 아닌 프로세서 그룹에 바인드합니다.	psrset(1M)

## ps 명령 사용

ps 명령을 사용하여 시스템에서 활성 프로세스의 상태를 확인하고, 프로세스에 대한 기술적 정보를 표시할 수 있습니다. 이 데이터는 프로세스 우선 순위 설정 방법 결정과 같은 관리 작업에 유용합니다.

사용하는 옵션에 따라 ps 명령은 다음 정보를 보고합니다.

- 프로세스의 현재 상태
- 프로세스 ID
- 상위 프로세스 ID
- 사용자 ID
- 예약 클래스
- 우선 순위
- 프로세스의 주소
- 사용된 메모리
- 사용된 CPU 시간

다음 표에서는 ps 명령으로 보고되는 몇 가지 필드를 설명합니다. 표시되는 옵션은 선택하는 옵션에 따라 달라집니다. 사용 가능한 모든 옵션에 대한 설명은 ps(1) 매뉴얼 페이지를 참조하십시오.

표 2-2 ps 보고서의 필드 요약

필드	설명
UID	프로세스 소유자의 유효 사용자 ID입니다.
PID	프로세스 ID입니다.
PPID	상위 프로세스 ID입니다.
C	예약을 위한 프로세서 사용률입니다. 이 필드는 -c 옵션이 사용되면 표시되지 않습니다.
CLS	실시간, 시스템 또는 시간 공유와 같이 프로세스가 속하는 예약 클래스입니다. 이 필드는 -c 옵션을 사용할 때만 포함됩니다.
PRI	커널 스레드의 예약 우선 순위입니다. 숫자가 높을수록 높은 우선 순위를 나타냅니다.
NI	예약 우선 순위에 영향을 주는 프로세스의 nice 번호입니다. 프로세스가 더 "nice"해지면 우선 순위가 낮음을 의미합니다.
ADDR	proc 구조의 속성입니다.
SZ	프로세스의 가상 주소 크기입니다.
WCHAN	프로세스가 일시 정지 상태인 이벤트 또는 잠금의 주소입니다.
STIME	프로세스의 시작 시간(시, 분, 초)입니다.
TTY	프로세스 또는 상위 프로세스가 시작된 터미널입니다. 물음표는 제어 터미널이 없음을 나타냅니다.
TIME	시작된 이후 프로세스에서 사용한 총 CPU 시간입니다.
CMD	프로세스를 생성한 명령입니다.

## /proc 파일 시스템 및 명령 사용

프로세스 명령을 사용하여 /proc 디렉토리에 나열된 프로세스에 대한 자세한 정보를 표시할 수 있습니다. 다음 표는 /proc 프로세스 명령을 나열합니다. 또한 /proc 디렉토리는 프로세스 파일 시스템(PROCFS)이라고 합니다. 활성 프로세스의 이미지는 여기에 해당 프로세스 ID 번호로 저장됩니다.

표 2-3 프로세스 명령(/proc)

프로세스 명령	설명
pcred	프로세스 자격 증명 정보를 표시합니다.

표 2-3 프로세스 명령(/proc) (계속)

프로세스 명령	설명
pfiles	프로세스에서 열린 파일에 대한 fstat 및 fcntl 정보를 보고합니다.
pflags	/proc 추적 플래그, 보류 중 신호/보류 신호 및 기타 상태 정보를 출력합니다.
pldd	프로세스에 연결된 동적 라이브러리를 나열합니다.
pmap	각 프로세스의 주소 공간 맵을 출력합니다.
psig	각 프로세스의 신호 작업 및 처리기를 나열합니다.
prun	각 프로세스를 시작합니다.
pstack	각 프로세스의 각 LWP에 대한 16진수+심볼릭 스택 추적을 출력합니다.
pstop	각 프로세스를 중지합니다.
ptime	미시상태 계산을 사용하여 프로세스 시간을 지정합니다.
ptree	프로세스를 포함하는 프로세스 트리를 표시합니다.
pwait	프로세스가 종료된 후 상태 정보를 표시합니다.
pwdx	프로세스에 대한 현재 작업 디렉토리를 표시합니다.

자세한 내용은 [proc\(1\)](#)을 참조하십시오.

프로세스 툴은 ps 명령의 일부 옵션과 유사하지만, 이러한 명령으로 제공되는 출력이 좀더 자세합니다.

일반적으로 프로세스 명령은 다음을 수행합니다.

- fstat 및 fcntl과 같은 프로세스, 작업 디렉토리 및 상위/하위 프로세스의 트리에 대한 추가 정보를 표시합니다.
- 사용자가 프로세스를 중지하거나 재개할 수 있도록 하여 프로세스에 대한 제어를 제공합니다.

## 프로세스 명령을 사용하여 프로세스 관리(/proc)

몇 가지 프로세스 명령을 사용하여 프로세스에 대한 자세한 기술적 정보를 표시하거나 활성 프로세스를 제어할 수 있습니다. 표 2-3은 일부 /proc 명령을 나열합니다.

프로세스가 무한 루프에 갇히게 되거나 프로세스를 실행하는데 너무 오래 걸리는 경우 프로세스를 중지(종료)할 수 있습니다. kill 또는 pkill 명령을 사용하여 프로세스를 중지하는 방법에 대한 자세한 내용은 2장, “시스템 프로세스 관리(작업)”를 참조하십시오.

/proc 파일 시스템은 상태 정보 및 제어 기능에 대한 추가 하위 디렉토리를 포함하는 디렉토리 계층입니다.

또한 /proc 파일 시스템은 프로세스 주소 공간의 개별 페이지에 대한 읽기 및 쓰기 권한을 재매핑하는 데 사용되는 xwatchpoint 기능도 제공합니다. 이 기능에는 제한 사항이 없으며 MT 안전입니다.

디버깅 툴은 /proc의 xwatchpoint 기능을 사용하도록 수정되어 전체 xwatchpoint 프로세스가 더 빨라졌습니다.

dbx 디버깅 툴을 사용하여 xwatchpoint를 설정할 때 다음 제한 사항이 제거되었습니다.

- SPARC 기반 시스템 등록 창으로 인한 스택의 로컬 변수에 대한 xwatchpoint 설정
- 다중 스레드 프로세스에 대한 xwatchpoint 설정

자세한 내용은 [proc\(4\)](#) 및 [mdb\(1\)](#) 매뉴얼 페이지를 참조하십시오.

## ▼ 프로세스를 나열하는 방법

- **ps 명령을 사용하여 시스템의 모든 프로세스를 나열합니다.**

```
$ ps [-efc]
```

ps     사용자의 로그인 세션과 연관된 프로세스만 표시합니다.

-ef     시스템에서 실행되는 모든 프로세스에 대한 전체 정보를 표시합니다.

-c     프로세스 스케줄 정보를 표시합니다.

### 예 2-1 프로세스 나열

다음 예는 옵션이 사용되지 않은 ps 명령의 출력을 보여줍니다.

```
$ ps
  PID TTY          TIME CMD
 1664 pts/4        0:06 csh
 2081 pts/4        0:00 ps
```

다음 예는 ps - ef 명령의 출력을 보여줍니다. 이 출력은 시스템이 부트될 때 실행된 첫 번째 프로세스가 sched(스왑퍼), init 프로세스, pageout 등으로 이어지고 있음을 나타냅니다.

```
$ ps -ef
UID    PID  PPID  C    STIME TTY          TIME CMD
root    0    0    0 18:04:04 ?        0:15 sched
root    5    0    0 18:04:03 ?        0:05 zpool-rpool
root    1    0    0 18:04:05 ?        0:00 /sbin/init
root    2    0    0 18:04:05 ?        0:00 pageout
root    3    0    0 18:04:05 ?        2:52 fsflush
root    6    0    0 18:04:05 ?        0:02 vmtasks
daemon 739  1    0 19:03:58 ?        0:00 /usr/lib/nfs/nfs4cbd
```

```

root      9      1      0 18:04:06 ?          0:14 /lib/svc/bin/svc.startd
root      11      1      0 18:04:06 ?          0:45 /lib/svc/bin/svc.configd
daemon   559     1      0 18:04:49 ?          0:00 /usr/sbin/rpcbind
netcfg   47      1      0 18:04:19 ?          0:01 /lib/inet/netcfgd
dladm    44      1      0 18:04:17 ?          0:00 /sbin/dlmgmt
netadm   51      1      0 18:04:22 ?          0:01 /lib/inet/ipmgmt
root     372     338    0 18:04:43 ?          0:00 /usr/lib/hal/hald-addon-cpufreq
root     67      1      0 18:04:30 ?          0:02 /lib/inet/in.mpathd
root     141     1      0 18:04:38 ?          0:00 /usr/lib/pfexecd
netadm   89      1      0 18:04:31 ?          0:03 /lib/inet/nwamd
root     602     1      0 18:04:50 ?          0:02 /usr/lib/inet/inetd start
root     131     1      0 18:04:35 ?          0:01 /sbin/dhccpagent
daemon   119     1      0 18:04:33 ?          0:00 /lib/crypto/kcfd
root     333     1      0 18:04:41 ?          0:07 /usr/lib/hal/hald --daemon=yes
root     370     338    0 18:04:43 ?          0:00 /usr/lib/hal/hald-addon-network-discovery
root     159     1      0 18:04:39 ?          0:00 /usr/lib/sysevent/syseventd
root     236     1      0 18:04:40 ?          0:00 /usr/lib/ldoms/drd
root     535     1      0 18:04:46 ?          0:09 /usr/sbin/nscd
root     305     1      0 18:04:40 ?          0:00 /usr/lib/zones/zonestatd
root     326     1      0 18:04:41 ?          0:03 /usr/lib/devfsadm/devfsadm
root     314     1      0 18:04:40 ?          0:00 /usr/lib/dbus-daemon --system

```

## ▼ 프로세스에 대한 정보를 표시하는 방법

- 1 추가 정보를 표시하고자 하는 프로세스의 프로세스 ID를 가져옵니다.

```
# pgrep process
```

여기서 *process*는 추가 정보를 표시하고자 하는 프로세스의 이름입니다.

프로세스 ID는 출력의 첫번째 열에 표시됩니다.

- 2 필요한 프로세스 정보를 표시합니다.

```
# /usr/bin/pcommand pid
```

*pcommand* 실행하고자 하는 (/proc) 명령입니다. 표 2-3은 이러한 명령을 나열하고 설명합니다.

*pid* 프로세스 ID를 식별합니다.

### 예 2-2 프로세스에 대한 정보 표시

다음 예는 프로세스 명령을 사용하여 cron 프로세스에 대한 추가 정보를 표시하는 방법을 보여줍니다.

```

# pgrep cron      1
4780
# pwdx 4780      2
4780: /var/spool/cron/atjobs

```

```
# ptree 4780      3
4780 /usr/sbin/cron
# pfiles 4780     4
4780: /usr/sbin/cron
Current rlimit: 256 file descriptors
 0: S_IFCHR mode:0666 dev:290,0 ino:6815752 uid:0 gid:3 rdev:13,2
    O_RDONLY|O_LARGEFILE
    /devices/pseudo/mm@0:null
 1: S_IFREG mode:0600 dev:32,128 ino:42054 uid:0 gid:0 size:9771
    O_WRONLY|O_APPEND|O_CREAT|O_LARGEFILE
    /var/cron/log
 2: S_IFREG mode:0600 dev:32,128 ino:42054 uid:0 gid:0 size:9771
    O_WRONLY|O_APPEND|O_CREAT|O_LARGEFILE
    /var/cron/log
 3: S_IFIFO mode:0600 dev:32,128 ino:42049 uid:0 gid:0 size:0
    O_RDWR|O_LARGEFILE
    /etc/cron.d/FIFO
 4: S_IFIFO mode:0000 dev:293,0 ino:4630 uid:0 gid:0 size:0
    O_RDWR|O_NONBLOCK
 5: S_IFIFO mode:0000 dev:293,0 ino:4630 uid:0 gid:0 size:0
    O_RDWR
```

1. cron 프로세스에 대한 프로세스 ID를 가져옵니다.
2. cron 프로세스에 대한 현재 작업 디렉토리를 표시합니다.
3. cron 프로세스를 포함하는 프로세스 트리를 표시합니다.
4. fstat 및 fcntl 정보를 표시합니다.

## ▼ 프로세스를 제어하는 방법

- 1 제어하고자 하는 프로세스의 프로세스 ID를 가져옵니다.

```
# pgrep process
```

여기서 *process*는 제어하고자 하는 프로세스의 이름입니다.

프로세스 ID는 출력의 첫번째 열에 표시됩니다.

- 2 적당한 프로세스 명령을 사용하여 프로세스를 제어합니다.

```
# /usr/bin/pcommand pid
```

*pcommand* 실행하고자 하는 프로세스(/proc) 명령입니다. 표 2-3은 이러한 명령을 나열하고 설명합니다.

*pid* 프로세스 ID를 식별합니다.

- 3 프로세스 상태를 확인합니다.

```
# ps -ef | grep pid
```

## 프로세스 종료(pkill, kill)

경우에 따라 프로세스를 중지(종료)해야 할 수 있습니다. 프로세스가 무한 루프에 있을 수 있습니다. 또는 완료되기 전에 중지하려고 하는 큰 작업을 시작했을 수 있습니다. 소유하고 있는 모든 프로세스를 종료할 수 있습니다. 슈퍼유저는 프로세스 ID가 0, 1, 2, 3 및 4인 프로세스를 제외하고 시스템의 모든 프로세스를 종료할 수 있습니다. 이러한 프로세스를 종료하면 시스템에 문제가 발생할 가능성이 큼니다.

자세한 내용은 `pgrep(1)` 및 `pkill(1)` 및 `kill(1)` 매뉴얼 페이지를 참조하십시오.

### ▼ 프로세스를 종료하는 방법(pkill)

- 1 다른 사용자의 프로세스를 종료하려면 `root` 역할을 가져야 합니다.

- 2 종료하고자 하는 프로세스의 프로세스 ID를 가져옵니다.

```
$ pgrep process
```

여기서 `process`는 종료하고자 하는 프로세스의 이름입니다.

예를 들면 다음과 같습니다.

```
$ pgrep netscape
587
566
```

프로세스 ID는 출력에 표시됩니다.

---

주 - Sun Ray에 대한 프로세스 정보를 가져오려면 다음 명령을 사용합니다.

```
# ps -fu user
```

이 명령은 모든 사용자 프로세스를 나열합니다.

```
# ps -fu user | grep process
```

이 명령은 사용자에 대한 특정 프로세스를 찾습니다.

- 3 프로세스를 종료합니다.

```
$ pkill [signal] process
```

**signal** pkill 명령줄 구문에 신호가 포함되지 않은 경우 사용되는 기본 신호는 `-15(SIGKILL)`입니다. pkill 명령과 함께 `-9` 신호(`SIGTERM`)를 사용하면 프로세스가 즉시 종료됩니다. 하지만 `-9` 신호는 데이터베이스 프로세스나 LDAP 서버 프로세스와 같은 특정 프로세스를 종료하는 데 사용하면 안 됩니다. 이 경우 데이터 손실이 발생할 수 있습니다.

**process** 중지할 프로세스의 이름입니다.

**참고** - `pkill` 명령을 사용하여 프로세스를 종료할 때 처음에는 신호 옵션을 포함시키지 말고 자체 명령만 사용하여 시도해 봅니다. `pkill` 명령을 `-9` 신호와 함께 사용하기 전에 몇 분 정도 기다린 후 프로세스가 종료되었는지 확인합니다.

**4 프로세스가 종료되었는지 확인합니다.**

```
$ pgrep process
```

종료한 프로세스는 `pgrep` 명령 출력에 더 이상 나열되지 않아야 합니다.

▼ **프로세스를 종료하는 방법(kill)**

**1 다른 사용자의 프로세스를 종료하려면 root 역할을 가져야 합니다.**

**2 종료하고자 하는 프로세스의 프로세스 ID를 가져옵니다.**

```
# ps -fu user
```

여기서 `user`는 프로세스를 표시하고자 하는 사용자입니다.

프로세스 ID는 출력의 첫번째 열에 표시됩니다.

**3 프로세스를 종료합니다.**

```
# kill [signal-number] pid
```

**signal** `kill` 명령줄 구문에 신호가 포함되지 않은 경우 사용되는 기본 신호는 `-15(SIGKILL)`입니다. `kill` 명령과 함께 `-9` 신호(`SIGTERM`)를 사용하면 프로세스가 즉시 종료됩니다. 하지만 `-9` 신호는 데이터베이스 프로세스나 LDAP 서버 프로세스와 같은 특정 프로세스를 종료하는 데 사용하면 안 됩니다. 이 경우 데이터 손실이 발생할 수 있습니다.

**pid** 종료하고자 하는 프로세스의 프로세스 ID입니다.

**참고** - `kill` 명령을 사용하여 프로세스를 중지할 때 처음에는 신호 옵션을 포함시키지 말고 자체 명령만 사용하여 시도해 봅니다. `kill` 명령을 `-9` 신호와 함께 사용하기 전에 몇 분 정도 기다린 후 프로세스가 종료되었는지 확인합니다.

**4 프로세스가 종료되었는지 확인합니다.**

```
$ pgrep pid
```

종료한 프로세스는 `pgrep` 명령 출력에 더 이상 나열되지 않아야 합니다.

**프로세스 디버깅(pargs, preap)**

`pargs` 명령 및 `preap` 명령은 프로세스 디버깅을 개선합니다. `pargs` 명령은 활성 프로세스 또는 코어 파일과 연관된 인수 및 환경 변수를 출력합니다. `preap` 명령은 `defunct`(좀비)

프로세스를 제거합니다. 좀비 프로세스는 상위 프로세스에서 요청한 종료 상태를 가지지 않습니다. 이러한 프로세스는 일반적으로 해를 끼치지 않지만 수가 많아지면 시스템 리소스를 소모할 수 있습니다. `pargs` 및 `preap` 명령을 사용하여 검사할 권한이 있는 프로세스를 검사할 수 있습니다. 슈퍼유저는 모든 프로세스를 검사할 수 있습니다.

`preap` 명령 사용에 대한 자세한 내용은 `preap(1)` 매뉴얼 페이지를 참조하십시오. `pargs` 명령 사용에 대한 자세한 내용은 `pargs(1)` 매뉴얼 페이지를 참조하십시오. 또한 `proc(1)` 매뉴얼 페이지를 참조하십시오.

### 예 2-3 프로세스 디버깅(`pargs`)

`pargs` 명령은 `ps` 명령으로 프로세스에 전달된 모든 인수를 표시할 수 없는 오래 지속되는 문제를 해결합니다. 다음 예는 `pgrep` 명령과 함께 `pargs` 명령을 사용하여 프로세스에 전달된 인수를 표시하는 방법을 보여줍니다.

```
# pargs 'pgrep ttymon'
579: /usr/lib/saf/ttymon -g -h -p system-name console login:
-T sun -d /dev/console -l
argv[0]: /usr/lib/saf/ttymon
argv[1]: -g
argv[2]: -h
argv[3]: -p
argv[4]: system-name console login:
argv[5]: -T
argv[6]: sun
argv[7]: -d
argv[8]: /dev/console
argv[9]: -l
argv[10]: console
argv[11]: -m
argv[12]: ldterm,ttcompat
548: /usr/lib/saf/ttymon
argv[0]: /usr/lib/saf/ttymon
```

다음 예는 `pargs -e` 명령을 사용하여 프로세스와 연관된 환경 변수를 표시하는 방법을 보여줍니다.

```
$ pargs -e 6763
6763: tcsh
envp[0]: DISPLAY=:0.0
```

## 프로세스 클래스 정보 관리

다음 목록은 시스템에서 구성할 수 있는 프로세스 예약 클래스를 나타냅니다. 또한 시간 공유 클래스에 대한 사용자가 우선 순위 범위도 포함되어 있습니다.

가능한 프로세스 예약 클래스는 다음과 같습니다.

- 공유(FSS)
- 고정(FX)
- 시스템(SYS)
- 대화식(IA)
- 실시간(RT)
- 시간 공유(TS)
  - 사용자 지정 우선 순위 범위는 -60부터 +60까지입니다.
  - 프로세스의 우선 순위는 상위 프로세스에서 상속됩니다. 이 우선 순위를 **사용자 모드 우선 순위**라고 합니다.
  - 시스템은 시간 공유 전달 매개변수 테이블에서 사용자 모드 우선 순위를 조회합니다. 그런 다음 시스템은 nice 또는 priocntl(사용자 지정) 우선 순위를 추가하고 0-59 범위가 **전역 우선 순위**를 생성하도록 합니다.

## 프로세스 클래스 정보 관리(작업 맵)

작업	설명	수행 방법
프로세스 클래스에 대한 기본 정보를 표시합니다.	priocntl -l 명령을 사용합니다. 프로세스 예약 클래스 및 우선 순위 범위를 표시합니다.	39 페이지 “프로세스 클래스에 대한 기본 정보를 표시하는 방법(priocntl)”
프로세스의 전역 우선 순위를 표시합니다.	ps -ecl 명령을 사용하여 프로세스의 전역 우선 순위를 표시합니다.	39 페이지 “프로세스의 전역 우선 순위를 표시하는 방법”
프로세스 우선 순위를 지정합니다.	priocntl -e -c 명령을 사용하여 지정된 우선 순위의 프로세스를 시작합니다.	41 페이지 “프로세스 우선 순위를 지정하는 방법(priocntl)”
시간 공유 프로세스의 예약 매개변수를 변경합니다.	priocntl -s -m 명령을 사용하여 시간 공유 프로세스의 예약 매개변수를 변경합니다.	42 페이지 “시간 공유 프로세스의 예약 매개변수를 변경하는 방법(priocntl)”
프로세스의 클래스를 변경합니다.	priocntl -s -c 명령을 사용하여 프로세스의 클래스를 변경합니다.	42 페이지 “프로세스의 클래스를 변경하는 방법(priocntl)”
프로세스의 우선 순위를 변경합니다.	/usr/bin/nice 명령을 적당한 옵션과 함께 사용하여 프로세스의 우선 순위를 낮추거나 높입니다.	44 페이지 “프로세스의 우선 순위를 변경하는 방법(nice)”

## 프로세스의 예약 우선 순위 변경(priocntl)

프로세스의 예약 우선 순위는 예약 정책에 따라 프로세스 스케줄러에서 지정한 우선 순위입니다. `dispadm` 명령은 기본 예약 정책을 나열합니다. 자세한 내용은 `dispadm(1M)` 매뉴얼 페이지를 참조하십시오.

`priocntl` 명령을 사용하여 프로세스를 우선 순위 클래스에 지정하고 프로세스 우선 순위를 관리할 수 있습니다. `priocntl` 명령을 사용하여 프로세스를 관리하는 방법에 대한 자세한 내용은 41 페이지 “프로세스 우선 순위를 지정하는 방법(`priocntl`)”을 참조하십시오.

### ▼ 프로세스 클래스에 대한 기본 정보를 표시하는 방법(`priocntl`)

- `priocntl -l` 명령을 사용하여 프로세스 예약 클래스 및 우선 순위 범위를 표시합니다.  
\$ `priocntl -l`

#### 예 2-4 프로세스 클래스에 대한 기본 정보 표시(`priocntl`)

다음 예는 `priocntl -l` 명령의 출력을 보여줍니다.

```
# priocntl -l
CONFIGURED CLASSES
=====

SYS (System Class)

TS (Time Sharing)
    Configured TS User Priority Range: -60 through 60

FX (Fixed priority)
    Configured FX User Priority Range: 0 through 60

IA (Interactive)
    Configured IA User Priority Range: -60 through 60
```

### ▼ 프로세스의 전역 우선 순위를 표시하는 방법

- `ps` 명령을 사용하여 프로세스의 전역 우선 순위를 표시합니다.  
\$ `ps -ecl`  
전역 우선 순위는 PRI 열에 나열됩니다.

### 예 2-5 프로세스의 전역 우선 순위 표시

다음 예에서는 ps -ecl 명령 출력을 보여줍니다. PRI 열의 값은 각 프로세스의 우선순위를 표시합니다.

```

$ ps -ecl
F S      UID    PID  PPID  CLS  PRI      ADDR      SZ      WCHAN  TTY      TIME CMD
1 T      0      0    0    SYS  96      ?      0      ?      ?      0:11 sched
1 S      0      5    0    SDC  99      ?      0      ? ?    ? ?    0:01 zpool-rp
0 S      0      1    0    TS   59      ?     688      ? ?    ? ?    0:00 init
1 S      0      2    0    SYS  98      ?      0      ? ?    ? ?    0:00 pageout
1 S      0      3    0    SYS  60      ?      0      ? ?    ? ?    2:31 fsflush
1 S      0      6    0    SDC  99      ?      0      ? ?    ? ?    0:00 vmtasks
0 S      16     56    1    TS   59      ?    1026      ? ?    ? ?    0:01 ipmgmt
0 S      0      9    1    TS   59      ?    3480      ? ?    ? ?    0:04 svc.star
0 S      0     11    1    TS   59      ?    3480      ? ?    ? ?    0:13 svc.conf
0 S      0     162    1    TS   59      ?     533      ? ?    ? ?    0:00 pfexecd
0 S      0    1738   1730  TS   59      ?     817      ? pts/ 1 0:00 bash
0 S      1     852    1    TS   59      ?     851      ? ?    ? ?    0:17 rpcbind
0 S      17     43    1    TS   59      ?    1096      ? ?    ? ?    0:01 netcfgd
0 S      15     47    1    TS   59      ?     765      ? ?    ? ?    0:00 dlmgmt
0 S      0      68    1    TS   59      ?     694      ? ?    ? ?    0:01 in.mpath
0 S      1    1220    1    FX   60      ?     682      ? ?    ? ?    0:00 nfs4cbd
0 S      16     89    1    TS   59      ?    1673      ? ?    ? ?    0:02 nward
0 S      0     146    1    TS   59      ?     629      ? ?    ? ?    0:01 dhcpgen
0 S      1     129    1    TS   59      ?    1843      ? ?    ? ?    0:00 kcf
0 S      1    1215    1    FX   60      ?     738      ? ?    ? ?    0:00 lockd
0 S      0     829    828  TS   59      ?     968      ? ?    ? ?    0:00 hald-run
0 S      0     361    1    TS   59      ?    1081      ? ?    ? ?    0:01 devfsadm
0 S      0     879    1    TS   59      ?    1166      ? ?    ? ?    0:01 inetd
0 0    119764 1773  880  TS   59      ?     557      ? cons ole 0:00 ps
0 S      0     844    829  TS   59      ?     996      ? ?    ? ?    0:00 hald-add
0 S      0     895    866  TS   59      ?     590      ? ?    ? ?    0:00 ttymon
0 S      0     840    1    TS   59      ?     495      ? ?    ? ?    0:00 cron
0 S      0     874    1    TS   59      ?     425      ? ?    ? ?    0:00 utmpd
0 S      0     1724   956  TS   59      ?    2215      ? ?    ? ?    0:00 sshd
0 S    119764 880     9    TS   59      ?     565      ? cons ole 0:00 csh
0 S      0     210    1    TS   59      ?    1622      ? ?    ? ?    0:00 sysevent
0 S      0     279    1    TS   59      ?     472      ? ?    ? ?    0:00 iscsid
0 S      1    1221    1    TS   59      ?    1349      ? ?    ? ?    0:00 nfsmapid
1 S      0     374    0    SDC  99      ?      0      ? ?    ? ?    0:00 zpool-us
0 S      0    1207    1    TS   59      ?    1063      ? ?    ? ?    0:00 rmvolmgr
0 S      0     828    1    TS   59      ?    1776      ? ?    ? ?    0:03 hald
0 S      0     853    829  TS   59      ?     896      ? ?    ? ?    0:02 hald-add
0 S      0     373    1    TS   59      ?     985      ? ?    ? ?    0:00 picld
0 S      0     299    1    TS   59      ?     836      ? ?    ? ?    0:00 dbus-dae
0 S    12524 1730  1725  TS   59      ?     452      ? pts/ 1 0:00 csh
0 S      0     370    1    TS   59      ?     574      ? ?    ? ?    0:00 powerd
0 S      0     264    1    FX   60      ?     637      ? ?    ? ?    0:00 zonestat
0 S      0     866     9    TS   59      ?     555      ? ?    ? ?    0:00 sac
0 S      0     851    829  TS   59      ?     998      ? ?    ? ?    0:00 hald-add
0 S    12524 1725  1724  TS   59      ?    2732      ? ?    ? ?    0:00 sshd
0 S      1    1211    1    TS   59      ?     783      ? ?    ? ?    0:00 statd
0 S      0    1046    1    TS   59      ?    1770      ? ?    ? ?    0:13 intrd
0 S      0     889    1    TS   59      ?    1063      ? ?    ? ?    0:00 syslogd
0 S      0    1209    1    TS   59      ?     792      ? ?    ? ?    0:00 in.ndpd
0 S      0    1188   1186  TS   59      ?     951      ? ?    ? ?    0:15 automoun
0 S      0    1172    829  TS   59      ?     725      ? ?    ? ?    0:00 hald-add
    
```

```

0 S      0 1186      1 TS 59      ? 692      ? ?      0:00 automoun
0 S     101 1739 1738 TS 59      ? 817      ? pts/ 1      0:00 bash
0 S      0 1199      1 TS 59      ? 1495     ? ?      0:02 sendmail
0 S      0 956        1 TS 59      ? 1729     ? ?      0:00 sshd
0 S     25 1192      1 TS 59      ? 1528     ? ?      0:00 sendmail
0 S      0 934        1 TS 59      ? 6897     ? ?      0:14 fmd
0 S      0 1131      1 TS 59      ? 1691     ? ?      0:07 nscd
0 S      1 1181      1 TS 59      ? 699      ? ?      0:00 ypbind

```

## ▼ 프로세스 우선 순위를 지정하는 방법(priocntl)

### 1 root 역할을 맡습니다.

**Oracle Solaris 11.1 관리: 보안 서비스의 “지정된 관리 권한을 사용하는 방법”을**  
참조하십시오.

### 2 지정된 우선 순위의 프로세스를 시작합니다.

```
# priocntl -e -c class -m user-limit -p pri command-name
```

-e 명령을 실행합니다.

-c class 프로세스를 실행할 클래스를 지정합니다. 유효한 클래스는 TS(시간 공유), RT(실시간), IA(대화식), FSS(공유) 및 FX(고정 우선 순위)입니다.

-m user-limit 이 옵션과 함께 -p 옵션을 사용할 경우 우선 순위를 높이거나 낮출 수 있는 최대 양도 지정됩니다.

-p pri command-name 실시간 스레드에 대해 RT 클래스의 상대 우선 순위를 지정할 수 있습니다. 시간 공유 프로세스의 경우 -p 옵션을 사용하여 -60부터 +60까지 사용자 지정 우선 순위를 지정할 수 있습니다.

### 3 프로세스 상태를 확인합니다.

```
# ps -ecl | grep command-name
```

## 예 2-6 프로세스 우선 순위 지정(priocntl)

다음 예는 가능한 가장 높은 사용자 지정 우선 순위로 find 명령을 시작하는 방법을 보여줍니다.

```
# priocntl -e -c TS -m 60 -p 60 find . -name core -print
# ps -ecl | grep find
```

## ▼ 시간 공유 프로세스의 예약 매개변수를 변경하는 방법(priocntl)

- 1 root 역할을 맡습니다.

**Oracle Solaris 11.1 관리: 보안 서비스의 “지정된 관리 권한을 사용하는 방법”을 참조하십시오.**

- 2 실행 중인 시간 공유 프로세스의 예약 매개변수를 변경합니다.

```
# priocntl -s -m user-limit [-p user-priority] -i idtype idlist
```

-s 사용자 우선 순위 범위에 대한 상한을 설정하고 현재 우선 순위를 변경할 수 있습니다.

-m user-limit -p 옵션을 사용할 때 우선 순위를 높이거나 낮출 수 있는 최대 양을 지정합니다.

-p user-priority 우선 순위를 지정할 수 있습니다.

-i xidtype xidlist xidtype 및 xidlist의 조합을 사용하여 프로세스를 식별합니다. xidtype은 프로세스 ID 또는 사용자 ID와 같은 ID의 유형을 지정합니다. xidlist를 사용하여 프로세스 ID 또는 사용자 ID의 목록을 식별합니다.

- 3 프로세스 상태를 확인합니다.

```
# ps -ecl | grep idlist
```

### 예 2-7 시간 공유 프로세스의 예약 매개변수 변경(priocntl)

다음 예는 500밀리초 간격, RT 클래스의 우선 순위 20 및 전역 우선 순위 120으로 명령을 실행하는 방법을 보여줍니다.

```
# priocntl -e -c RT -m 500 -p 20 myprog
# ps -ecl | grep myprog
```

## ▼ 프로세스의 클래스를 변경하는 방법(priocntl)

- 1 (옵션) root 역할을 맡습니다.

**Oracle Solaris 11.1 관리: 보안 서비스의 “지정된 관리 권한을 사용하는 방법”을 참조하십시오.**

- 2 프로세스의 클래스를 변경합니다.

```
# priocntl -s -c class -i idtype idlist
```

- s 사용자 우선 순위 범위에 대한 상한을 설정하고 현재 우선 순위를 변경할 수 있습니다.
- c class 프로세스를 변경하는 클래스(시간 공유의 경우 TS 또는 실시간의 경우 RT)를 지정합니다.
- i idtype idlist *xidtype* 및 *xidlist*의 조합을 사용하여 프로세스를 식별합니다. *xidtype*은 프로세스 ID 또는 사용자 ID와 같은 ID의 유형을 지정합니다. *xidlist*를 사용하여 프로세스 ID 또는 사용자 ID의 목록을 식별합니다.

---

주 - 실시간 프로세스에서 또는 실시간 프로세스로 프로세스를 변경하려면 root 역할을 맡거나 실시간 셸에서 작업 중이어야 합니다. root 역할을 맡은 사용자가 사용자 프로세스를 실시간 클래스로 변경할 경우 이후 사용자는 `prionctl -s` 명령을 사용하여 실시간 예약 매개변수를 변경할 수 없습니다.

---

### 3 프로세스 상태를 확인합니다.

```
# ps -ecl | grep idlist
```

## 예 2-8 프로세스의 클래스 변경(prionctl)

다음 예는 사용자 15249에 속한 모든 프로세스를 실시간 프로세스로 변경하는 방법을 보여줍니다.

```
# prionctl -s -c RT -i uid 15249
# ps -ecl | grep 15249
```

## 시간 공유 프로세스의 우선 순위 변경(nice)

`nice` 명령은 이전 릴리스에 대한 역호환성을 위해서만 지원됩니다. `prionctl` 명령이 프로세스 관리에서 더욱 많은 유연성을 제공합니다.

프로세스의 우선 순위는 예약 클래스의 정책 및 해당 *nice* 번호로 결정됩니다. 각 시간 공유 프로세스에는 전역 우선 순위가 있습니다. 전역 우선 순위는 `nice` 또는 `prionctl` 명령의 영향을 받을 수 있는 사용자 지정 우선 순위와 시스템 계산 우선 순위를 더하여 계산됩니다.

프로세스의 실행 우선 순위 번호는 운영 체제에서 지정합니다. 우선 순위 번호는 프로세스의 예약 클래스, 사용한 CPU 시간 및 시간 공유 프로세스의 경우 해당 *nice* 번호를 포함한 여러 요소로 결정됩니다.

각 시간 공유 프로세스는 상위 프로세스에서 상속된 기본 *nice* 번호로 시작됩니다. *nice* 번호는 `ps` 보고서의 NI 열에 표시됩니다.

사용자는 사용자 지정 우선 순위를 높여 프로세스의 우선 순위를 낮출 수 있습니다. 하지만 슈퍼유저만 nice 번호를 낮추어 프로세스의 우선 순위를 높일 수 있습니다. 이 제한 사항은 사용자가 자신의 프로세스 우선 순위를 높여 CPU를 독점하지 못하도록 합니다.

nice 번호의 범위는 0부터 +39까지이며, 0이 가장 높은 우선 순위를 나타냅니다. 각 시간 공유 프로세스에 대한 nice 기본값은 20입니다. 표준 버전인 /usr/bin/nice와 C 셸 내장 명령의 두 가지 버전의 명령을 사용할 수 있습니다.

## ▼ 프로세스의 우선 순위를 변경하는 방법(nice)

이 절차에 따라 사용자가 프로세스의 우선 순위를 낮출 수 있습니다. 하지만 root 역할은 프로세스의 우선 순위를 높이거나 낮출 수 있습니다.

- 1 프로세스의 우선 순위를 사용자 또는 슈퍼유저로 변경할지 결정합니다. 그런 다음, 다음 중 하나를 선택합니다.

- 사용자는 단계 2의 예를 따라 명령의 우선 순위를 낮춥니다.
- 슈퍼유저는 단계 3의 예를 따라 명령의 우선 순위를 높이거나 낮춥니다.

- 2 사용자는 nice 번호를 높여 명령의 우선 순위를 낮춥니다.

다음 nice 명령은 nice 번호를 5단위씩 높여 낮은 우선 순위의 *command-name*을 실행합니다.

```
$ /usr/bin/nice -5 command-name
```

위의 명령에서 마이너스 기호는 뒤따르는 옵션을 지정합니다. 이 명령은 다음과 같이 지정할 수도 있습니다.

```
$ /usr/bin/nice -n 5 command-name
```

다음 nice 명령은 nice 번호를 기본 증분값인 10단위씩 높이지만 최대값인 39를 초과하지 않도록 하여 *command-name*의 우선 순위를 낮춥니다.

```
$ /usr/bin/nice command-name
```

- 3 슈퍼유저는 nice 번호를 변경하여 명령의 우선 순위를 높이거나 낮출 수 있습니다.

다음 nice 명령은 nice 번호를 10단위씩 낮추지만 최소값인 0 미만이 되지 않도록 하여 *command-name*의 우선 순위를 높입니다.

```
# /usr/bin/nice --10 command-name
```

위의 명령에서 첫번째 마이너스 기호는 뒤따르는 옵션을 지정합니다. 두번째 마이너스 기호는 음수를 나타냅니다.

다음 nice 명령은 nice 번호를 5단위씩 높이지만 최대값인 39를 초과하지 않도록 하여 *command-name*의 우선 순위를 낮춥니다.

```
# /usr/bin/nice -5 command-name
```

참조 자세한 내용은 [nice\(1\)](#) 매뉴얼 페이지를 참조하십시오.

## 시스템 프로세스의 문제 해결

다음은 일반적으로 발생할 수 있는 문제에 대한 몇 가지 유용한 정보입니다.

- 같은 사용자가 소유하고 있는 여러 동일 작업을 찾습니다. 이 문제는 작업이 완료될 때까지 기다리지 않고 많은 백그라운드 작업을 시작하는 실행 스크립트로 인해 발생할 수 있습니다.
- 많은 양의 CPU 시간이 누적된 프로세스를 찾습니다. `ps` 출력에서 `TIME` 필드를 확인하여 이 문제를 파악할 수 있습니다. 프로세스가 무한 루프에 있을 수 있습니다.
- 너무 높은 우선 순위로 실행되는 프로세스를 찾습니다. `ps -c` 명령을 사용하여 각 프로세스의 예약 클래스를 표시하는 `CLS` 필드를 확인합니다. 실시간(RT) 프로세스로 실행되는 프로세스는 CPU를 독점할 수 있습니다. 또는 높은 nice 번호의 시간 공유(TS) 프로세스를 찾습니다. 슈퍼유저 권한을 가진 사용자가 프로세스의 우선 순위를 높였을 수 있습니다. 시스템 관리자는 nice 명령을 사용하여 우선 순위를 낮출 수 있습니다.
- runaway 프로세스를 찾습니다. runaway 프로세스는 갈수록 더 많은 CPU 시간을 사용합니다. 프로세스가 시작된 시간(STIME)을 확인하고 잠시 동안 CPU 시간 누적(TIME)을 살펴 이 문제를 파악할 수 있습니다.



## 시스템 성능 모니터링(작업)

---

컴퓨터 또는 네트워크로부터 뛰어난 성능을 얻는 것은 시스템 관리의 중요한 부분입니다. 이 장에서는 사용자가 관리하는 컴퓨터 시스템의 성능을 관리하는 데 기여하는 몇 가지 요소에 대한 개요를 제공합니다. 특히 이 장에서는 `vmstat`, `iostat`, `df` 및 `sar` 명령을 사용하여 시스템 성능을 모니터링하기 위한 절차를 설명합니다.

다음은 이 장에서 다루는 정보를 나열한 것입니다.

- 47 페이지 “시스템 성능 작업의 위치”
- 48 페이지 “시스템 성능 및 시스템 리소스”
- 48 페이지 “프로세스 및 시스템 성능”
- 50 페이지 “시스템 성능 모니터링 정보”
- 51 페이지 “시스템 성능 정보 표시”
- 59 페이지 “시스템 작업 모니터링”

### 시스템 성능 작업의 위치

시스템 성능 작업	자세한 정보
프로세스 관리	2 장, “시스템 프로세스 관리(작업)”
시스템 성능 모니터	3 장, “시스템 성능 모니터링(작업)”
조정 가능 매개변수 변경	<b>Oracle Solaris 11.1 조정 가능 매개변수 참조 설명서</b>
시스템 성능 작업 관리	<b>Oracle Solaris 관리: Oracle Solaris Zones, Oracle Solaris 10 Zones 및 리소스 관리</b> 의 2 장, “프로젝트 및 작업(개요)”
FX 및 FS 스케줄러를 사용하여 프로세스 관리	<b>Oracle Solaris 관리: Oracle Solaris Zones, Oracle Solaris 10 Zones 및 리소스 관리</b> 의 8 장, “FSS(Fair Share Scheduler)(개요)”

## 시스템 성능 및 시스템 리소스

컴퓨터 시스템의 성능은 시스템에서 리소스를 사용하고 할당하는 방식에 따라 달라집니다. 정상적인 조건에서 시스템이 어떻게 작동하는지 알 수 있도록 시스템의 성능을 정기적으로 모니터링하십시오. 문제를 예상하고 문제 발생 시 문제를 인식할 수 있어야 합니다.

성능에 영향을 미치는 시스템 리소스는 다음 표에 설명되어 있습니다.

시스템 리소스	설명
중앙 처리 장치(CPU)	CPU는 메모리에서 명령어를 불러온 다음 실행하여 명령어를 처리합니다.
입/출력(I/O) 장치	I/O 장치는 컴퓨터에서 정보를 전송하거나 컴퓨터로 정보를 전송합니다. 터미널 및 키보드, 디스크 드라이브 또는 프린터가 이러한 장치일 수 있습니다.
메모리	물리적(또는 주) 메모리는 시스템의 RAM(Random Access Memory) 양입니다.

3 장, “시스템 성능 모니터링(작업)”에는 시스템의 작업 및 성능에 대한 통계를 표시하는 도구가 설명되어 있습니다.

## 프로세스 및 시스템 성능

다음 표는 프로세스와 관련된 용어에 대해 설명합니다.

표 3-1 프로세스 용어

용어	설명
프로세스	임의의 시스템 작업입니다. 사용자가 시스템을 부트하거나, 명령을 실행하거나, 응용 프로그램을 시작할 때마다 시스템에서 하나 이상의 프로세스가 활성화됩니다.
LWP(Lightweight Process)	가상 CPU 또는 실행 리소스입니다. LWP는 예약 클래스 및 우선 순위를 기준으로 사용 가능한 CPU 리소스를 사용하도록 커널에 의해 예약됩니다. LWP에는 커널 스레드와 LWP가 포함되어 있습니다. 커널 스레드에는 항상 메모리에 있어야 하는 정보가 포함되어 있습니다. LWP에는 스왑 가능한 정보가 포함되어 있습니다.
응용 프로그램 스레드	사용자의 주소 공간에서 개별적으로 실행할 수 있는 별도의 스택을 포함하는 일련의 명령어입니다. 응용 프로그램 스레드는 LWP를 기반으로 멀티플렉싱될 수 있습니다.

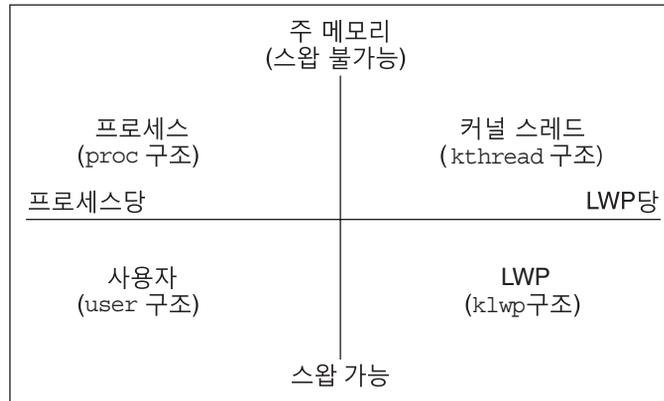
프로세스는 여러 개의 LWP와 여러 개의 응용 프로그램 스레드로 구성될 수 있습니다. 커널은 SunOS 환경의 예약 엔티티인 커널 스레드 구조를 예약합니다. 다양한 프로세스 구조는 다음 표에 설명되어 있습니다.

표 3-2 프로세스 구조

구조	설명
proc	전체 프로세스에 속하는 정보가 포함되어 있으며 항상 주 메모리에 있어야 합니다.
kthread	하나의 LWP에 속하는 정보가 포함되어 있으며 항상 주 메모리에 있어야 합니다.
user	스왑 가능한 "프로세스별" 정보가 포함되어 있습니다.
klwp	스왑 가능한 "LWP 프로세스별" 정보가 포함되어 있습니다.

다음 그림은 이러한 프로세스 구조 간의 관계를 보여줍니다.

그림 3-1 프로세스 구조 간의 관계



대부분의 프로세스 리소스는 프로세스의 모든 스레드에서 액세스할 수 있습니다. 거의 모든 프로세스 가상 메모리는 공유 메모리입니다. 한 스레드에 의해 공유 데이터가 변경되면 프로세스의 다른 스레드에서 이 변경 사항을 사용할 수 있습니다.

# 시스템 성능 모니터링 정보

컴퓨터가 실행 중인 동안 운영 체제의 카운터는 다양한 시스템 작업을 추적하기 위해 증분됩니다.

추적되는 시스템 작업은 다음과 같습니다.

- 중앙 처리 장치(CPU) 사용량
- 버퍼 사용량
- 디스크 및 테이프 입/출력(I/O) 작업
- 터미널 장치 작업
- 시스템 호출 작업
- 컨텍스트 전환
- 파일 액세스
- 대기열 작업
- 커널 테이블
- 프로세스 간 통신
- 페이지징
- 사용 가능한 메모리 및 스왑 공간
- 커널 메모리 할당(KMA)

## 모니터링 도구

Oracle Solaris 소프트웨어는 시스템이 작업을 수행하는 방식을 추적하는 데 유용한 여러 가지 도구를 제공합니다.

표 3-3 성능 모니터링 도구

명령	설명	자세한 정보
cpustat 및 cputrack 명령	CPU 성능 카운터를 사용하여 시스템 또는 프로세스의 성능을 모니터링합니다.	cpustat(1M) 및 cputrack(1)
netstat 및 nfsstat 명령	네트워크 성능에 대한 정보를 표시합니다.	netstat(1M) 및 nfsstat(1M)
ps 및 prstat 명령	활성 프로세스에 대한 정보를 표시합니다.	2 장, “시스템 프로세스 관리(작업)”
sar 및 sadc 명령	시스템 작업 데이터를 수집하고 보고합니다.	3 장, “시스템 성능 모니터링(작업)”
swap 명령	시스템의 사용 가능한 스왑 공간에 대한 정보를 표시합니다.	Oracle Solaris 11.1 관리: 장치 및 파일 시스템의 16 장, “추가 스왑 공간 구성(작업)”

표 3-3 성능 모니터링 도구 (계속)

명령	설명	자세한 정보
vmstat 및 iostat 명령	가상 메모리 통계, 디스크 사용량 및 CPU 작업 등의 시스템 작업 데이터를 요약합니다.	3 장, “시스템 성능 모니터링(작업)”
kstat 및 mpstat 명령	시스템에서 사용 가능한 커널 통계 또는 kstats를 확인하고, 명령줄에 지정된 기준과 일치하는 통계를 보고합니다. mpstat 명령은 프로세서 통계를 표 형식으로 보고합니다.	kstat(1M) 및 mpstat(1M) 매뉴얼 페이지

## 시스템 성능 정보 표시

이 절에서는 시스템 성능 정보 표시를 모니터하기 위한 작업에 대해 설명합니다.

### 시스템 성능 정보 표시(작업 맵)

작업	설명	수행 방법
가상 메모리 통계를 표시합니다.	vmstat 명령을 사용하여 가상 메모리 통계를 수집합니다.	53 페이지 “가상 메모리 통계를 표시하는 방법(vmstat)”
시스템 이벤트 정보를 표시합니다.	vmstat 명령을 -s 옵션과 함께 사용하여 시스템 이벤트 정보를 표시합니다.	53 페이지 “시스템 이벤트 정보를 표시하는 방법(vmstat -s)”
스와핑 통계를 표시합니다.	vmstat 명령을 -S 옵션과 함께 사용하여 스와핑 통계를 표시합니다.	54 페이지 “스와핑 통계를 표시하는 방법(vmstat -S)”
장치당 인터럽트를 표시합니다.	vmstat 명령을 -i 옵션과 함께 사용하여 장치당 인터럽트의 수를 표시합니다.	54 페이지 “장치당 인터럽트를 표시하는 방법(vmstat -i)”
디스크 사용량을 표시합니다.	iostat 명령을 사용하여 디스크 입출력 통계를 보고합니다.	55 페이지 “디스크 사용량 정보를 표시하는 방법(iostat)”
확장 디스크 통계를 표시합니다.	iostat 명령을 -xtc 옵션과 함께 사용하여 확장 디스크 통계를 표시합니다.	56 페이지 “확장 디스크 통계를 표시하는 방법(iostat -xtc)”
디스크 공간 정보를 표시합니다.	df -k 명령은 디스크 공간 정보를 킬로바이트 단위로 표시합니다.	57 페이지 “디스크 공간 정보를 표시하는 방법(df -k)”

## 가상 메모리 통계 표시(vmstat)

vmstat 명령을 사용하여 CPU 로드, 페이징, 컨텍스트 전환 수, 장치 인터럽트 및 시스템 호출과 같은 시스템 이벤트에 대한 가상 메모리 통계 및 정보를 보고할 수 있습니다. vmstat 명령은 스와핑, 캐시 비우기 및 인터럽트에 대한 통계도 표시할 수 있습니다.

표 3-4 vmstat 명령의 출력

범주	필드 이름	설명
procs		다음에 대해 보고합니다.
	r	전달 대기열의 커널 스레드 수
	b	리소스를 대기 중인 차단된 커널 스레드 수
	w	처리 중인 리소스 완료를 대기 중인 스왑 아웃된 LWP 수
memory		실제 메모리 및 가상 메모리의 사용에 대해 보고합니다.
	swap	사용 가능한 스왑 공간
	free	해제 목록의 크기
page		페이지 폴트 및 페이지 작업을 초 단위로 보고합니다.
	re	재확보된 페이지
	mf	작은 결함 및 큰 결함
	pi	페이징인(킬로바이트)
	po	페이징아웃(킬로바이트)
	fr	해제됨(킬로바이트)
	de	최근 스왑 인된 프로세스에서 필요한 예상 메모리
	sr	현재 사용되고 있지 않은 page 데몬으로 스캔된 페이지. sr이 0이 아니면 page 데몬이 실행된 것입니다.
disk	최대 4개 디스크의 데이터에 대한 초당 디스크 작업 수를 보고합니다.	
faults		초당 트랩/인터럽트 비율을 보고합니다.
	in	초당 인터럽트
	sy	초당 시스템 호출
	cs	CPU 컨텍스트 전환 비율
cpu		CPU 시간 사용을 보고합니다.
	us	사용자 시간

표 3-4 vmstat 명령의 출력 (계속)

범주	필드 이름	설명
	sy	시스템 시간
	id	유휴 시간

이 명령에 대한 자세한 내용은 `vmstat(1M)` 매뉴얼 페이지를 참조하십시오.

### ▼ 가상 메모리 통계를 표시하는 방법(vmstat)

- `vmstat` 명령을 시간 간격(초)과 함께 사용하여 가상 메모리 통계를 수집합니다.

```
$ vmstat n
```

여기서 `n`은 보고 사이의 간격(초)입니다.

#### 예 3-1 가상 메모리 통계 표시

다음 예는 5초 간격으로 수집된 통계의 `vmstat` 표시를 보여줍니다.

```
$ vmstat 5
kthr      memory          page        disk        faults        cpu
 r  b  w    swap free  re  mf  pi  po  fr  de  sr  dd  f0  s1  --   in  sy   cs  us  sy  id
0  0  0  863160 365680  0   3  1  0  0  0  0  0  0  0  0  406 378  209  1  0  99
0  0  0  765640 208568  0  36  0  0  0  0  0  0  0  0  0  479 4445 1378  3  3  94
0  0  0  765640 208568  0   0  0  0  0  0  0  0  0  0  0  423  214  235  0  0 100
0  0  0  765712 208640  0   0  0  0  0  0  0  3  0  0  0  412  158  181  0  0 100
0  0  0  765832 208760  0   0  0  0  0  0  0  0  0  0  0  402  157  179  0  0 100
0  0  0  765832 208760  0   0  0  0  0  0  0  0  0  0  0  403  153  182  0  0 100
0  0  0  765832 208760  0   0  0  0  0  0  0  0  0  0  0  402  168  177  0  0 100
0  0  0  765832 208760  0   0  0  0  0  0  0  0  0  0  0  402  153  178  0  0 100
0  0  0  765832 208760  0  18  0  0  0  0  0  0  0  0  0  407  165  186  0  0 100
```

### ▼ 시스템 이벤트 정보를 표시하는 방법(vmstat -s)

- `vmstat -s` 명령을 실행하여 시스템이 마지막으로 부트된 이후 얼마나 많은 시스템 이벤트가 발생했는지 표시합니다.

```
$ vmstat -s
  0 swap ins
  0 swap outs
  0 pages swapped in
  0 pages swapped out
522586 total address trans. faults taken
17006 page ins
  25 page outs
23361 pages paged in
  28 pages paged out
45594 total reclaims
45592 reclaims from free list
  0 micro (hat) faults
```

```

522586 minor (as) faults
16189 major faults
98241 copy-on-write faults
137280 zero fill page faults
45052 pages examined by the clock daemon
    0 revolutions of the clock hand
    26 pages freed by the clock daemon
2857 forks
    78 vforks
1647 execs
34673885 cpu context switches
65943468 device interrupts
711250 traps
63957605 system calls
3523925 total name lookups (cache hits 99%)
    92590 user    cpu
    65952 system cpu
16085832 idle   cpu
    7450 wait   cpu
    
```

### ▼ 스와핑 통계를 표시하는 방법(vmstat -S)

- **vmstat -S**를 실행하여 스와핑 통계를 표시합니다.

```

$ vmstat -S
kthr      memory          page            disk          faults        cpu
 r  b  w  swap free  si  so pi po fr de sr dd f0 s1 --  in sy  cs us sy id
  0  0  0 862608 364792  0  0  1  0  0  0  0  0  0  0  406 394 213  1  0 99
    
```

스와핑 통계 필드는 다음 목록에 설명되어 있습니다. 기타 필드에 대한 설명은 [표 3-4](#)를 참조하십시오.

si    초당 스왑 인된 평균 LWP 수  
so    스왑 아웃된 프로세스 수

주 - vmstat 명령은 si 및 so 필드의 출력을 자릅니다. 스왑 통계의 좀더 정확한 계산을 표시하려면 sar 명령을 사용합니다.

### ▼ 장치당 인터럽트를 표시하는 방법(vmstat -i)

- **vmstat -i** 명령을 실행하여 장치당 인터럽트 수를 표시합니다.

#### 예 3-2 장치당 인터럽트 표시

다음 예는 vmstat -i 명령의 출력을 보여줍니다.

```

$ vmstat -i
interrupt          total          rate
-----
cLock              52163269      100
    
```

esp0	2600077	4
zsc0	25341	0
zsc1	48917	0
cgsixc0	459	0
lec0	400882	0
fdc0	14	0
bppc0	0	0
audiocs0	0	0
-----		
Total	55238959	105

## 디스크 사용률 정보 표시(iostat)

iostat 명령을 사용하여 디스크 입출력에 대한 통계를 보고하고 처리량, 사용률, 대기열 길이, 트랜잭션 비율 및 서비스 시간에 대한 측정 결과를 표시합니다. 이 명령에 대한 자세한 내용은 [iostat\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

### ▼ 디스크 사용률 정보를 표시하는 방법(iostat)

- iostat 명령을 시간 간격(초)과 함께 사용하여 디스크 사용률 정보를 표시할 수 있습니다.

```
$ iostat 5
      tty          fd0          sd3          nfs1          nfs31          cpu
tin tout kps tps serv kps tps serv kps tps serv kps tps serv us sy wt id
  0    1  0  0  410   3  0  29   0  0   9   3  0  47  4  2  0  94
```

출력의 첫 라인은 시스템이 마지막으로 부트된 이후의 통계를 보여줍니다. 이후 각 라인은 간격 통계를 보여줍니다. 기본값은 터미널(tty), 디스크(fd 및 sd) 및 CPU(cpu)에 대한 통계를 보여주는 것입니다.

### 예 3-3 디스크 사용률 정보 표시

다음 예는 5초마다 수집된 디스크 통계를 보여줍니다.

```
$ iostat 5
      tty          sd0          sd6          nfs1          nfs49          cpu
tin tout kps tps serv kps tps serv kps tps serv kps tps serv us sy wt id
  0    0  1  0  49   0  0   0   0  0   0   0  0  15  0  0  0 100
  0   47  0  0   0   0  0   0   0  0   0   0  0   0  0  0  0 100
  0   16  0  0   0   0  0   0   0  0   0   0  0   0  0  0  0 100
  0   16  0  0   0   0  0   0   0  0   0   0  0   0  0  0  0 100
  0   16 44  6 132   0  0   0   0  0   0   0  0   0  0  0  1  99
  0   16  0  0   0   0  0   0   0  0   0   0  0   0  0  0  0 100
  0   16  0  0   0   0  0   0   0  0   0   0  0   0  0  0  0 100
  0   16  0  0   0   0  0   0   0  0   0   0  0   0  0  0  0 100
  0   16  0  0   0   0  0   0   0  0   0   0  0   0  0  0  0 100
  0   16  0  0   0   0  0   0   0  0   0   0  0   0  0  0  0 100
  0   16  3  1  23   0  0   0   0  0   0   0  0   0  0  0  1  99
  0   16  0  0   0   0  0   0   0  0   0   0  0   0  0  0  0 100
  0   16  0  0   0   0  0   0   0  0   0   0  0   0  0  0  0 100
```

0 16 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 100

다음 표는 `iostat n` 명령 출력의 필드를 설명합니다.

장치 유형	필드 이름	설명
터미널	장치 유형	
	tin	터미널 입력 대기열의 문자 수
	tout	터미널 출력 대기열의 문자 수
디스크	장치 유형	
	bps	초당 블록
	tps	초당 트랜잭션
	serv	평균 서비스 시간(밀리초)
CPU	장치 유형	
	us	사용자 모드
	sy	시스템 모드
	wt	I/O 대기 중
	id	유휴

### ▼ 확장 디스크 통계를 표시하는 방법(`iostat -xtc`)

- `iostat -xtc` 명령을 실행하여 확장 디스크 통계를 표시합니다.

```
$ iostat -xtc
                extended device statistics
device         r/s   w/s   kr/s   kw/s  wait  actv  svc_t  %w  %b   tty          cpu
               0.0   0.0   0.0   0.0  0.0  0.0   0.0  0  0   tin tout  us sy wt id
fd0             0.0   0.0   0.0   0.0  0.0  0.0   0.0  0  0    0  0  0  0  0 100
sd0             0.0   0.0   0.4   0.4  0.0  0.0   49.5  0  0
sd6             0.0   0.0   0.0   0.0  0.0  0.0    0.0  0  0
nfs1            0.0   0.0   0.0   0.0  0.0  0.0    0.0  0  0
nfs49           0.0   0.0   0.0   0.0  0.0  0.0   15.1  0  0
nfs53           0.0   0.0   0.4   0.0  0.0  0.0   24.5  0  0
nfs54           0.0   0.0   0.0   0.0  0.0  0.0    6.3  0  0
nfs55           0.0   0.0   0.0   0.0  0.0  0.0    4.9  0  0
```

`iostat -xtc` 명령은 각 디스크에 대해 한 라인에 출력을 표시합니다. 출력 필드는 다음 목록에 설명되어 있습니다.

- r/s      초당 읽기
- w/s      초당 쓰기
- kr/s     초당 읽은 킬로바이트

kw/s	초당 쓴 킬로바이트
wait	서비스를 대기 중인 평균 트랜잭션 수(대기열 길이)
actv	서비스 중인 평균 활성 트랜잭션 수
svc_t	평균 서비스 시간(밀리초)
%w	대기열이 비어 있지 않은 시간(백분율)
%b	디스크를 사용 중인 시간(백분율)

## 디스크 공간 통계 표시(df)

df 명령을 사용하여 각 마운트된 디스크에서 사용 가능한 디스크 공간을 표시합니다. 보고 통계에서는 총 사용 가능 공간보다 10%를 허용하므로 df로 보고되는 **사용 가능** 디스크 공간에는 전체 용량의 90%만 반영됩니다. 이 **헤드 공간**은 일반적으로 보다 좋은 성능을 위해 비워 둡니다.

df 명령으로 실제로 보고되는 디스크 공간의 비율은 사용 가능 공간으로 나눈 사용된 공간입니다.

파일 시스템이 90% 용량을 초과할 경우 cp 명령을 사용하여 가득 차지 않은 디스크로 파일을 전송할 수 있습니다. tar 또는 cpio 명령을 사용하여 테이프로 파일을 전송할 수도 있습니다. 또는 파일을 제거할 수 있습니다.

이 명령에 대한 자세한 내용은 [df\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

### ▼ 디스크 공간 정보를 표시하는 방법(df -k)

- df -k 명령을 사용하여 디스크 공간 정보를 킬로바이트 단위로 표시합니다.

```
$ df -k
Filesystem          kbytes    used   avail capacity  Mounted on
/dev/dsk/c0t3d0s0  192807    40231  133296    24%      /
```

### 예 3-4 파일 시스템 정보 표시

다음 예는 df -k 명령의 출력을 보여줍니다.

```
$ df -k
Filesystem          1024-blocks    Used Available Capacity  Mounted on
rpool/ROOT/solaris-161  191987712    6004395  140577816     5%      /
/devices              0              0         0      0%      /devices
/dev                  0              0         0      0%      /dev
ctfs                  0              0         0      0%      /system/contract
proc                  0              0         0      0%      /proc
mnttab                0              0         0      0%      /etc/mnttab
```

swap	4184236	496	4183740	1%	/system/volatile
objfs	0	0	0	0%	/system/object
sharefs	0	0	0	0%	/etc/dfs/sharetab
/usr/lib/libc/libc_hwcapi.so.1	146582211	6004395	140577816	5%	/lib/libc.so.1
fd	0	0	0	0%	/dev/fd
swap	4183784	60	4183724	1%	/tmp
rpool/export	191987712	35	140577816	1%	/export
rpool/export/home	191987712	32	140577816	1%	/export/home
rpool/export/home/123	191987712	13108813	140577816	9%	/export/home/123
rpool/export/repo	191987712	11187204	140577816	8%	/export/repo
rpool/export/repo2010_11	191987712	31	140577816	1%	/export/repo2010_11
rpool	191987712	5238974	140577816	4%	/rpool
/export/home/123	153686630	13108813	140577816	9%	/home/123

다음 표는 df -k 명령의 출력을 설명합니다.

필드 이름	설명
kbytes	파일 시스템에서 총 사용 가능한 공간 크기
used	사용된 공간의 양
avail	사용 가능한 공간의 양
capacity	사용된 공간의 양(총 용량의 백분율)
mounted on	마운트 지점

### 예 3-5 df 명령을 옵션 없이 사용하여 파일 시스템 정보 표시

다음 예와 같이 df 명령을 피연산자 또는 옵션 없이 사용할 경우 마운트된 모든 파일 시스템이 보고됩니다.

```
$ df
/ (rpool/ROOT/solaris):100715496 blocks 100715496 files
/devices (/devices ): 0 blocks 0 files
/dev (/dev ): 0 blocks 0 files
/system/contract (ctfs ): 0 blocks 2147483601 files
/proc (proc ): 0 blocks 29946 files
/etc/mnttab (mnttab ): 0 blocks 0 files
/system/volatile (swap ):42257568 blocks 2276112 files
/system/object (objfs ): 0 blocks 2147483441 files
/etc/dfs/sharetab (sharefs ): 0 blocks 2147483646 files
/dev/fd (fd ): 0 blocks 0 files
/tmp (swap ):42257568 blocks 2276112 files
/export (rpool/export ):100715496 blocks 100715496 files
/export/home (rpool/export/home ):100715496 blocks 100715496 files
/export/home/admin (rpool/export/home/admin):100715496 blocks 100715496 files
/rpool (rpool ):100715496 blocks 100715496 files
/export/repo2010_11(rpool/export/repo2010_11):281155639 blocks 281155639 files
/rpool (rpool ):281155639 blocks 281155639 files
```

# 시스템 작업 모니터링

이 절에서는 시스템 작업을 모니터링하기 위한 작업에 대해 설명합니다.

## 시스템 작업 모니터링(작업 맵)

작업	설명	수행 방법
파일 액세스를 확인합니다.	sar 명령을 -a 옵션과 함께 사용하여 파일 액세스 작업 상태를 표시합니다.	60 페이지 “파일 액세스를 확인하는 방법(sar -a)”
버퍼 작업을 확인합니다.	sar 명령을 -b 옵션과 함께 사용하여 버퍼 작업 통계를 표시합니다.	61 페이지 “버퍼 작업을 확인하는 방법(sar -b)”
시스템 호출 통계를 확인합니다.	sar 명령을 -c 옵션과 함께 사용하여 시스템 호출 통계를 표시합니다.	63 페이지 “시스템 호출 통계를 확인하는 방법(sar -c)”
디스크 작업을 확인합니다.	sar 명령을 -d 옵션과 함께 사용하여 디스크 작업을 확인합니다.	64 페이지 “디스크 작업을 확인하는 방법(sar -d)”
페이지 아웃 및 메모리를 확인합니다.	sar 명령을 -g 옵션과 함께 사용하여 페이지 아웃 메모리 해제 작업을 표시합니다.	65 페이지 “페이지 아웃 및 메모리를 확인하는 방법(sar -g)”
커널 메모리 할당을 확인합니다.	커널 메모리 할당(KMA)을 통해 커널 부속 시스템에서 필요에 따라 메모리를 할당하고 해제할 수 있습니다. sar 명령을 -k 옵션과 함께 사용하여 KMA를 확인합니다.	67 페이지 “커널 메모리 할당을 확인하는 방법(sar -k)”
프로세스간 통신을 확인합니다.	sar 명령을 -m 옵션과 함께 사용하여 프로세스간 통신 작업을 보고합니다.	68 페이지 “프로세스간 통신을 확인하는 방법(sar -m)”
페이지 인 작업을 확인합니다.	sar 명령을 -p 옵션과 함께 사용하여 페이지 인 작업을 보고합니다.	69 페이지 “페이지 인 작업을 확인하는 방법(sar -p)”
대기열 작업을 확인합니다.	sar 명령을 -q 옵션과 함께 사용하여 다음을 확인합니다. <ul style="list-style-type: none"> <li>■ 대기열이 채워진 동안의 평균 대기열 길이</li> <li>■ 대기열이 채워진 시간(백분율)</li> </ul>	70 페이지 “대기열 작업을 확인하는 방법(sar -q)”
사용되지 않은 메모리를 확인합니다.	sar 명령을 -r 옵션과 함께 사용하여 현재 사용되고 있는 메모리 페이지 및 스왑 파일 디스크 블록 수를 보고합니다.	71 페이지 “사용되지 않은 메모리를 확인하는 방법(sar -r)”
CPU 사용률을 확인합니다.	sar 명령을 -u 옵션과 함께 사용하여 CPU 사용률 통계를 표시합니다.	72 페이지 “CPU 사용률을 확인하는 방법(sar -u)”

작업	설명	수행 방법
시스템 테이블 상태를 확인합니다.	<p>sar 명령을 -v 옵션과 함께 사용하여 다음 시스템 테이블에 대한 상태를 보고합니다.</p> <ul style="list-style-type: none"> <li>■ 프로세스</li> <li>■ Inode</li> <li>■ 파일</li> <li>■ 공유 메모리 레코드</li> </ul>	73 페이지 “시스템 테이블 상태를 확인하는 방법(sar -v)”
스와핑 작업을 확인합니다.	sar 명령을 -w 옵션과 함께 사용하여 스와핑 작업을 확인합니다.	74 페이지 “스와핑 작업을 확인하는 방법(sar -w)”
터미널 작업을 확인합니다.	sar 명령을 -y 옵션과 함께 사용하여 터미널 장치 작업을 모니터링합니다.	75 페이지 “터미널 작업을 확인하는 방법(sar -y)”
전체 시스템 성능을 확인합니다.	sar -A 명령은 모든 옵션의 통계를 표시하여 전체 시스템 성능 정보를 제공합니다.	76 페이지 “전체 시스템 성능을 확인하는 방법(sar -A)”
자동 데이터 수집을 설정합니다.	<p>시스템에서 자동으로 데이터를 수집하도록 설정하고 sar 명령을 실행하려면 다음을 수행합니다.</p> <ul style="list-style-type: none"> <li>■ svcadm enable system/sar:default 명령을 실행합니다.</li> <li>■ /var/spool/cron/crontabs/sys 파일을 편집합니다.</li> </ul>	79 페이지 “자동 데이터 수집을 설정하는 방법”

## 시스템 작업 모니터링(sar)

sar 명령을 사용하여 다음 작업을 수행합니다.

- 시스템 작업에 대한 데이터를 구성하고 확인합니다.
- 특별 요청에 따라 시스템 작업 데이터에 액세스합니다.
- 자동 보고서를 생성하여 시스템 성능을 측정 및 모니터링하고, 특별 요청 보고서를 생성하여 특정 성능 문제를 파악합니다. sar 명령이 시스템에서 실행되도록 설정하는 방법 및 이러한 도구에 대한 자세한 내용은 76 페이지 “자동으로 시스템 작업 데이터 수집(sar)”을 참조하십시오.

이 명령에 대한 자세한 내용은 sar(1) 매뉴얼 페이지를 참조하십시오.

### ▼ 파일 액세스를 확인하는 방법(sar -a)

- sar -a 명령을 사용하여 파일 액세스 작업 통계를 표시합니다.

```
$ sar -a
```

```
SunOS t2k-brm-24 5.10 Generic_144500-10 sun4v ...
```

```
00:00:00  iget/s namei/s dirbk/s
01:00:00      0      3      0
02:00:00      0      3      0
03:00:00      0      3      0
04:00:00      0      3      0
05:00:00      0      3      0
06:00:00      0      3      0
07:00:00      0      3      0
08:00:00      0      3      0
08:20:01      0      3      0
08:40:00      0      3      0
09:00:00      0      3      0
09:20:01      0     10      0
09:40:01      0      1      0
10:00:02      0      5      0

Average      0      4      0
```

다음 목록은 sar -a 명령으로 보고되는 필드 이름 및 운영 체제 루틴을 설명합니다.

**iget/s** 디렉토리 이름 조회 캐시(DNLC)에 없는 inode에 대한 요청 수입입니다.

**namei/s** 초당 파일 시스템 경로 검색 수입입니다. namei가 DNLC에서 디렉토리 이름을 찾지 못할 경우 iget을 호출하여 파일이나 디렉토리에 대한 inode를 가져옵니다. 따라서 대부분의 igets는 DNLC 실패의 결과입니다.

**dirbk/s** 초당 수행된 디렉토리 블록 읽기 수입입니다.

이러한 운영 체제 루틴에 대해 보고되는 값이 클수록 커널이 사용자 파일에 액세스하는데 소비하는 시간이 많아집니다. 시간의 양에는 얼마나 많은 프로그램 및 응용 프로그램이 파일 시스템을 사용하고 있는지 반영됩니다. -a 옵션은 응용 프로그램이 얼마나 디스크 종속적인지 보는 데 유용합니다.

## ▼ 버퍼 작업을 확인하는 방법(sar -b)

- sar -b 명령을 사용하여 버퍼 작업 통계를 표시합니다.

버퍼는 메타 데이터를 캐시에 저장하는 데 사용됩니다. 메타 데이터에는 inode, 실린더 그룹 블록 및 간접 블록이 포함됩니다.

```
$ sar -b
00:00:00  bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
01:00:00      0      0      100      0      0      55      0      0
```

### 예 3-6 버퍼 작업 확인(sar -b)

sar -b 명령 출력의 다음 예는 %rcache 및 %wcache 버퍼가 성능 저하를 유발하고 있지 않음을 보여줍니다. 모든 데이터가 허용 가능한 한도 내에 있습니다.

```
$ sar -b

SunOS t2k-brm-24 5.10 Generic_144500-10 sun4v ...
```

00:00:04	bread/s	lread/s	%rcache	bwrit/s	lwrit/s	%wcache	pread/s	pwrit/s
01:00:00	0	0	100	0	0	94	0	0
02:00:01	0	0	100	0	0	94	0	0
03:00:00	0	0	100	0	0	92	0	0
04:00:00	0	1	100	0	1	94	0	0
05:00:00	0	0	100	0	0	93	0	0
06:00:00	0	0	100	0	0	93	0	0
07:00:00	0	0	100	0	0	93	0	0
08:00:00	0	0	100	0	0	93	0	0
08:20:00	0	1	100	0	1	94	0	0
08:40:01	0	1	100	0	1	93	0	0
09:00:00	0	1	100	0	1	93	0	0
09:20:00	0	1	100	0	1	93	0	0
09:40:00	0	2	100	0	1	89	0	0
10:00:00	0	9	100	0	5	92	0	0
10:20:00	0	0	100	0	0	68	0	0
10:40:00	0	1	98	0	1	70	0	0
11:00:00	0	1	100	0	1	75	0	0
Average	0	1	100	0	1	91	0	0

다음 표는 -b 옵션으로 표시되는 버퍼 작업을 표시합니다.

필드 이름	설명
bread/s	디스크에서 버퍼 캐시로 제출된 초당 평균 읽기 수
lread/s	버퍼 캐시에서 초당 평균 논리적 읽기 수
%rcache	버퍼 캐시에서 발견된 논리적 읽기의 비율(100% - bread/s/lread/s 비율)
bwrit/s	버퍼 캐시에서 디스크에 쓰여진 초당 평균 물리적 블록 수(512바이트)
lwrit/s	버퍼 캐시로 초당 평균 논리적 쓰기 수
%wcache	버퍼 캐시에서 발견된 논리적 쓰기의 비율(100% - bwrit/s/lwrit/s의 비율)
pread/s	문자 장치 인터페이스를 사용하는 초당 평균 물리적 읽기 수
pwrit/s	문자 장치 인터페이스를 사용하는 초당 평균 물리적 쓰기 요청 수

가장 중요한 항목은 캐시 적중률 %rcache 및 %wcache입니다. 이러한 항목은 시스템 버퍼링의 유효성을 측정합니다. %rcache가 90% 아래로 떨어지거나 %wcache가 65% 아래로 떨어질 경우 버퍼 공간을 늘려 성능을 향상시킬 수 있습니다.

## ▼ 시스템 호출 통계를 확인하는 방법(sar -c)

- sar -c 명령을 사용하여 시스템 호출 통계를 표시합니다.

```
$ sar -c
00:00:00 scall/s sread/s swrit/s fork/s exec/s rchar/s wchar/s
01:00:00      38      2      2  0.00  0.00   149   120
```

### 예 3-7 시스템 호출 통계 확인(sar -c)

다음 예는 sar -c 명령의 출력을 보여줍니다.

```
$ sar -c

SunOS balmy 5.10 Generic_144500-10 sun4v ...
00:00:04 scall/s sread/s swrit/s fork/s exec/s rchar/s wchar/s
01:00:00      89      14      9  0.01  0.00   2906   2394
02:00:01      89      14      9  0.01  0.00   2905   2393
03:00:00      89      14      9  0.01  0.00   2908   2393
04:00:00      90      14      9  0.01  0.00   2912   2393
05:00:00      89      14      9  0.01  0.00   2905   2393
06:00:00      89      14      9  0.01  0.00   2905   2393
07:00:00      89      14      9  0.01  0.00   2905   2393
08:00:00      89      14      9  0.01  0.00   2906   2393
08:20:00      90      14      9  0.01  0.01   2914   2395
08:40:01      90      14      9  0.01  0.00   2914   2396
09:00:00      90      14      9  0.01  0.01   2915   2396
09:20:00      90      14      9  0.01  0.01   2915   2396
09:40:00     880     207     156  0.08  0.08  26671   9290
10:00:00    2020     530     322  0.14  0.13  57675  36393
10:20:00     853     129      75  0.02  0.01  10500   8594
10:40:00    2061     524     450  0.08  0.08 579217 567072
11:00:00    1658     404     350  0.07  0.06 1152916 1144203

Average      302      66      49  0.02  0.01  57842  55544
```

다음 표는 -c 옵션으로 보고되는 시스템 호출 범주를 설명합니다. 일반적으로 읽기 및 쓰기가 전체 시스템 호출의 약 절반을 차지합니다. 하지만 비율은 시스템에서 수행되는 작업에 따라 크게 달라집니다.

필드 이름	설명
scall/s	초당 모든 유형의 시스템 호출 수입니다. 일반적으로 4-6명의 시스템에서 초당 약 30입니다.
sread/s	초당 read 시스템 호출 수입니다.
swrit/s	초당 write 시스템 호출 수입니다.
fork/s	초당 fork 시스템 호출 수입니다. 4-6명의 시스템에서 초당 약 0.5입니다. 셸 스크립트가 실행되는 경우 이 수는 늘어납니다.
exec/s	초당 exec 시스템 호출 수입니다. fork/s로 나눈 exec/s가 3보다 클 경우 비효율적인 PATH 변수를 찾으십시오.

필드 이름	설명
rchar/s	초당 read 시스템 호출로 전송된 문자 수(바이트)입니다.
wchar/s	초당 write 시스템 호출로 전송된 문자 수(바이트)입니다.

### ▼ 디스크 작업을 확인하는 방법(sar -d)

- **sar -d** 명령을 사용하여 디스크 작업 통계를 표시합니다.

```
$ sar -d
00:00:00 device      %busy  avque  r+w/s  blks/s  await  avserv
```

#### 예 3-8 디스크 작업 확인

이 약식 예는 sar -d 명령의 출력을 보여줍니다.

```
$ sar -d
SunOS balmy 5.10 Generic_144500-10 sun4v  ...
12:36:32 device      %busy  avque  r+w/s  blks/s  await  avserv
12:40:01 dad1        15     0.7   26     399    18.1   10.0
      dad1,a    15     0.7   26     398    18.1   10.0
      dad1,b     0     0.0    0      1      1.0    3.0
      dad1,c     0     0.0    0      0      0.0    0.0
      dad1,h     0     0.0    0      0      0.0    6.0
      fd0        0     0.0    0      0      0.0    0.0
      nfs1       0     0.0    0      0      0.0    0.0
      nfs2       1     0.0    1     12     0.0   13.2
      nfs3       0     0.0    0      2     0.0    1.9
      nfs4       0     0.0    0      0     0.0    7.0
      nfs5       0     0.0    0      0     0.0   57.1
      nfs6       1     0.0    6    125     4.3    3.2
      nfs7       0     0.0    0      0     0.0    6.0
      sd1        0     0.0    0      0     0.0    5.4
      ohci0,bu   0     0.0    0      0     0.0    0.0
      ohci0,ct   0     0.0    0      0     0.0    0.0
      ohci0,in   0     0.0    7      0     0.0    0.0
      ohci0,is   0     0.0    0      0     0.0    0.0
      ohci0,to   0     0.0    7      0     0.0    0.0
```

다음 표는 -d 옵션으로 보고되는 디스크 장치 작업을 설명합니다.

필드 이름	설명
장치	모니터링되는 디스크 장치의 이름입니다.
%busy	장치가 전송 요청을 서비스하는 시간입니다.
avque	장치가 전송 요청을 서비스하는 시간 동안의 평균 요청 수입니다.

필드 이름	설명
r+w/s	장치로 초당 읽기 및 쓰기 전송 수입니다.
blks/s	장치로 전송된 초당 512바이트 블록 수입니다.
await	전송 요청이 대기열에서 유류 상태로 대기하는 평균 시간(밀리초)입니다. 이 시간은 대기열이 채워진 경우에만 측정됩니다.
avserv	장치에서 전송 요청을 완료하는 평균 시간(밀리초)입니다. 디스크의 경우 이 값에는 탐색 시간, 회전 지연 시간 및 데이터 전송 시간이 포함됩니다.

대기열 길이 및 대기 시간은 대기열이 채워진 경우에만 측정됩니다. %busy가 작은 경우 큰 대기열 및 서비스 시간은 시스템에서 변경된 블록이 디스크에 즉시 쓰여지도록 주기적인 작업을 수행하고 있음을 나타냅니다.

## ▼ 페이지 아웃 및 메모리를 확인하는 방법(sar -g)

- sar -g 명령을 사용하여 평균 페이지 아웃 및 메모리 해제 작업을 표시합니다.

```
$ sar -g
00:00:00 pgout/s ppgout/s pgfree/s pgscan/s %ufs_ipf
01:00:00 0.00 0.00 0.00 0.00 0.00
```

sar -g 명령으로 표시되는 출력은 더 많은 메모리가 필요한지 여부를 알 수 있는 좋은 지표입니다. ps -elf 명령을 사용하여 page 데몬에서 사용되는 주기 수를 표시합니다. pgfree/s 및 pgscan/s 필드에 대한 높은 값과 함께 높은 주기 수는 메모리 부족을 나타냅니다.

또한 sar -g 명령은 inode가 너무 빨리 재사용되어 재사용 가능한 페이지 유실을 일으키는 지 여부를 보여줍니다.

### 예 3-9 페이지 아웃 및 메모리 확인(sar -g)

다음 예는 sar -g 명령의 출력을 보여줍니다.

```
$ sar -g
SunOS balmy 5.10 Generic_144500-10 sun4v ...
00:00:00 pgout/s ppgout/s pgfree/s pgscan/s %ufs_ipf
01:00:00 0.00 0.00 0.00 0.00 0.00
02:00:00 0.01 0.01 0.01 0.00 0.00
03:00:00 0.00 0.00 0.00 0.00 0.00
04:00:00 0.00 0.00 0.00 0.00 0.00
05:00:00 0.00 0.00 0.00 0.00 0.00
06:00:00 0.00 0.00 0.00 0.00 0.00
07:00:00 0.00 0.00 0.00 0.00 0.00
08:00:00 0.00 0.00 0.00 0.00 0.00
```

08:20:01	0.00	0.00	0.00	0.00	0.00
08:40:00	0.00	0.00	0.00	0.00	0.00
09:00:00	0.00	0.00	0.00	0.00	0.00
09:20:01	0.05	0.52	1.62	10.16	0.00
09:40:01	0.03	0.44	1.47	4.77	0.00
10:00:02	0.13	2.00	4.38	12.28	0.00
10:20:03	0.37	4.68	12.26	33.80	0.00
Average	0.02	0.25	0.64	1.97	0.00

다음 표는 -g 옵션의 출력을 설명합니다.

필드 이름	설명
pgout/s	초당 페이지 아웃 요청 수입니다.
ppgout/s	초당 페이지 아웃된 실제 페이지 수입니다. 단일 페이지 아웃 요청에는 여러 페이지의 페이지 아웃이 포함될 수 있습니다.
pgfree/s	여유 목록에 있는 초당 페이지 수입니다.
pgscan/s	page 데몬으로 스캔된 초당 페이지 수입니다. 이 값이 높을 경우 page 데몬이 여유 메모리를 확인하는 데 많은 시간을 소비하고 있는 것입니다. 이 상황은 더 많은 메모리가 필요함을 나타냅니다.
%ufs_ipf	연관된 재사용 가능한 페이지가 있는 해제 목록에서 iget으로 가져간 ufs inode의 백분율입니다. 이러한 페이지는 비워지며 프로세스에서 재확보할 수 없습니다. 따라서 이 필드는 페이지 비우기에서 igets의 백분율을 나타냅니다. 높은 값은 inode의 여유 목록이 페이지 제한적이고, ufs inode의 수를 늘려야 함을 나타냅니다.

## 커널 메모리 할당 확인

KMA를 통해 커널 부속 시스템에서 필요에 따라 메모리를 할당하고 해제할 수 있습니다.

부하가 가장 많은 시간대에 필요할 것으로 예상되는 최대 메모리 양을 정적으로 할당하는 대신 KMA는 메모리에 대한 요청을 세 범주로 나눕니다.

- 작음(256바이트 미만)
- 큼(512바이트- 4킬로바이트)
- 매우 큼(4킬로바이트 이상)

KMA는 작은 요청과 큰 요청을 충족하기 위해 두 메모리 풀을 유지합니다. 매우 큰 요청은 시스템 페이지 할당기에서 메모리를 할당하여 충족됩니다.

KMA 리소스를 사용하는 드라이버 또는 STREAMS를 쓰는 데 사용되는 시스템을 확인하는 경우 sar -k 명령이 유용할 수 있습니다. 그렇지 않은 경우 제공되는 정보가 필요하지 않을 수 있습니다. KMA 리소스를 사용하지만 종료하기 전에 명시적으로

리소스를 반환하지 않는 드라이버나 모듈은 메모리 누수를 유발할 수 있습니다. 메모리 누수로 인해 시간에 따라 KMA로 할당되는 메모리의 양이 늘어나게 됩니다. 따라서 sar -k 명령의 alloc 필드가 시간에 따라 계속 늘어나면 메모리 누수가 있을 수 있습니다. 또 하나의 메모리 누수 지표는 요청 실패입니다. 이 문제가 발생할 경우 메모리 누수로 인해 KMA에서 메모리를 확보하고 할당할 수 없게 됩니다.

메모리 누수가 발생할 것으로 보이면 KMA에서 메모리를 요청하고 반환하지 않은 드라이버나 STREAMS를 확인해야 합니다.

## ▼ 커널 메모리 할당을 확인하는 방법(sar -k)

- sar -k 명령을 사용하여 커널 메모리 할당기(KMA)의 다음 작업을 보고합니다.

```
$ sar -k
00:00:00 sml_mem alloc fail lg_mem alloc fail ovsz_alloc fail
01:00:00 2523136 1866512 0 18939904 14762364 0 360448 0
02:00:02 2523136 1861724 0 18939904 14778748 0 360448 0
```

### 예 3-10 커널 메모리 할당 확인(sar -k)

다음은 sar -k 출력의 약식 예입니다.

```
$ sar -k
SunOS balmy 5.10 Generic_144500-10 sun4v ...
00:00:04 sml_mem alloc fail lg_mem alloc fail ovsz_alloc fail
01:00:00 6119744 4852865 0 60243968 54334808 156 9666560 0
02:00:01 6119744 4853057 0 60243968 54336088 156 9666560 0
03:00:00 6119744 4853297 0 60243968 54335760 156 9666560 0
04:00:00 6119744 4857673 0 60252160 54375280 156 9666560 0
05:00:00 6119744 4858097 0 60252160 54376240 156 9666560 0
06:00:00 6119744 4858289 0 60252160 54375608 156 9666560 0
07:00:00 6119744 4858793 0 60252160 54442424 156 9666560 0
08:00:00 6119744 4858985 0 60252160 54474552 156 9666560 0
08:20:00 6119744 4858169 0 60252160 54377400 156 9666560 0
08:40:01 6119744 4857345 0 60252160 54376880 156 9666560 0
09:00:00 6119744 4859433 0 60252160 54539752 156 9666560 0
09:20:00 6119744 4858633 0 60252160 54410920 156 9666560 0
09:40:00 6127936 5262064 0 60530688 55619816 156 9666560 0
10:00:00 6545728 5823137 0 62996480 58391136 156 9666560 0
10:20:00 6545728 5758997 0 62996480 57907400 156 9666560 0
10:40:00 6734144 6035759 0 64389120 59743064 156 10493952 0
11:00:00 6996288 6394872 0 65437696 60935936 156 10493952 0

Average 6258044 5150556 0 61138340 55609004 156 9763900 0
```

다음 표는 -k 옵션의 출력을 설명합니다.

필드 이름	설명
sml_mem	KMA가 작은 메모리 요청 풀에서 사용할 수 있는 메모리의 양(바이트)입니다. 이 풀에서 작은 요청은 256바이트 미만입니다.
alloc	KMA가 작은 메모리 요청 풀에서 작은 메모리 요청에 할당된 메모리의 양(바이트)입니다.
fail	작은 메모리 양에 대해 실패한 요청 수입니다.
lg_mem	KMA가 큰 메모리 요청 풀에서 사용할 수 있는 메모리의 양(바이트)입니다. 이 풀에서 큰 요청은 512바이트-4킬로바이트입니다.
alloc	KMA가 큰 메모리 요청 풀에서 큰 메모리 요청에 할당된 메모리의 양(바이트)입니다.
fail	큰 메모리 양에 대해 실패한 요청 수입니다.
ovsz_alloc	4킬로바이트보다 큰 요청인 매우 큰 요청에 대해 할당된 메모리의 양입니다. 이러한 요청은 페이지 할당기로 충족됩니다. 따라서 풀이 없습니다.
fail	매우 큰 메모리 양에 대해 실패한 요청 수입니다.

## ▼ 프로세스간 통신을 확인하는 방법(sar -m)

- sar -m 명령을 사용하여 프로세스간 통신 작업을 보고합니다.

```
$ sar -m
00:00:00      msg/s   sema/s
01:00:00      0.00     0.00
```

메시지나 세마포어를 사용하는 응용 프로그램을 실행하지 않는다면 이러한 수치는 대개 0.00입니다.

다음 목록은 -m 옵션의 출력을 설명합니다.

```
msg/s      초당 메시지 작업(송수신) 수
sema/s     초당 세마포어 작업 수
```

### 예 3-11 프로세스간 통신 확인(sar -m)

다음 약식 예는 sar -m 명령의 출력을 보여줍니다.

```
$ sar -m
SunOS balmy 5.10 Generic_144500-10 sun4v      ...
00:00:00      msg/s   sema/s
01:00:00      0.00     0.00
```

```

02:00:02    0.00    0.00
03:00:00    0.00    0.00
04:00:00    0.00    0.00
05:00:01    0.00    0.00
06:00:00    0.00    0.00

Average      0.00    0.00
    
```

### ▼ 페이지 인 작업을 확인하는 방법(sar -p)

- sar -p 명령을 사용하여 보호 및 변환 폴트를 포함하는 페이지 인 작업을 보고합니다.

```

$ sar -p
00:00:00 atch/s pgin/s ppgin/s pflt/s vflt/s slock/s
01:00:00  0.07  0.00  0.00  0.21  0.39  0.00
    
```

### 예 3-12 페이지 인 작업 확인(sar -p)

다음 예는 sar -p 명령의 출력을 보여줍니다.

```

$ sar -p

SunOS balmy 5.10 Generic_144500-10 sun4v    ...

00:00:04 atch/s pgin/s ppgin/s pflt/s vflt/s slock/s
01:00:00  0.09  0.00  0.00  0.78  2.02  0.00
02:00:01  0.08  0.00  0.00  0.78  2.02  0.00
03:00:00  0.09  0.00  0.00  0.81  2.07  0.00
04:00:00  0.11  0.01  0.01  0.86  2.18  0.00
05:00:00  0.08  0.00  0.00  0.78  2.02  0.00
06:00:00  0.09  0.00  0.00  0.78  2.02  0.00
07:00:00  0.08  0.00  0.00  0.78  2.02  0.00
08:00:00  0.09  0.00  0.00  0.78  2.02  0.00
08:20:00  0.11  0.00  0.00  0.87  2.24  0.00
08:40:01  0.13  0.00  0.00  0.90  2.29  0.00
09:00:00  0.11  0.00  0.00  0.88  2.24  0.00
09:20:00  0.10  0.00  0.00  0.88  2.24  0.00
09:40:00  2.91  1.80  2.38  4.61 17.62  0.00
10:00:00  2.74  2.03  3.08  8.17 21.76  0.00
10:20:00  0.16  0.04  0.04  1.92  2.96  0.00
10:40:00  2.10  2.50  3.42  6.62 16.51  0.00
11:00:00  3.36  0.87  1.35  3.92 15.12  0.00

Average   0.42  0.22  0.31  1.45  4.00  0.00
    
```

다음 표는 -p 옵션으로 보고된 통계를 설명합니다.

필드 이름	설명
atch/s	현재 메모리에 있는 페이지를 확보하여 충족된 초당 페이지 폴트 수입니다(초당 첨부). 인스턴스에는 여유 목록에서 잘못된 페이지 확보 및 현재 다른 프로세스에서 사용되고 있는 텍스트 페이지 공유가 포함됩니다. 동일한 프로그램 텍스트에 액세스하고 있는 둘 이상의 프로세스를 예로 들 수 있습니다.
pgin/s	파일 시스템이 페이지 인 요청을 수신하는 초당 횟수입니다.
ppgin/s	초당 페이지 인된 페이지 수입니다. 소프트 잠금 요청(slock/s 참조) 또는 큰 블록 크기와 같은 단일 페이지 인 요청에는 여러 페이지의 페이지 인이 포함될 수 있습니다.
pflt/s	보호 오류의 페이지 폴트 수입니다. 보호 결함 인스턴스는 페이지 및 “쓰기 시 복사”에 대한 잘못된 액세스를 나타냅니다. 일반적으로 이 수는 주로 “쓰기 시 복사”로 구성됩니다.
vflt/s	초당 주소 변환 페이지 폴트 수입니다. 이러한 결함을 유효성 결함이라고 합니다. 유효성 결함은 유효한 프로세스 테이블 항목이 해당 가상 주소에 대해 존재하지 않을 때 발생합니다.
slock/s	물리적 I/O가 필요한 소프트웨어 잠금 요청으로 인해 발생하는 초당 결함 수입니다. 예를 들어, 소프트웨어 잠금 요청은 디스크에서 메모리로 데이터를 전송할 때 발생할 수 있습니다. 시스템이 데이터를 수신해야 하는 페이지를 잠그므로 다른 페이지에서 페이지를 확보하고 사용할 수 없습니다.

## ▼ 대기열 작업을 확인하는 방법(sar -q)

### ● sar -q 명령을 사용하여 다음 정보를 보고합니다.

- 대기열이 채워진 동안의 평균 대기열 길이
- 대기열이 채워진 시간(백분율)

```
$ sar -q
00:00:00 runq-sz %runocc swpq-sz %swpocc
```

다음 목록은 -q 옵션의 출력을 설명합니다.

**runq-sz** 실행을 위해 CPU를 대기 중인 메모리의 커널 스레드 수입니다. 일반적으로 이 값은 2보다 작아야 합니다. 지속적으로 높은 값은 시스템이 CPU 제한적임을 나타냅니다.

**%runocc** 전달 대기열이 채워진 시간의 백분율입니다.

**swpq-sz** 교체되어 나온 프로세스의 평균 개수입니다.

**%swpocc** 프로세스가 교체되어 나온 시간의 백분율입니다.

### 예 3-13 대기열 작업 확인

다음 예는 sar -q 명령의 출력을 보여줍니다. %runocc 값이 높고(90% 이상) runq-sz 값이 2보다 클 경우, CPU 부하가 높고 응답 속도가 떨어집니다. 이 경우 허용할 만한 시스템 응답 속도를 얻으려면 추가 CPU가 필요할 수 있습니다.

```
# sar -q
SunOS balmy 5.10 Generic_144500-10 sun4v    ...

00:00:00 runq-sz %runocc swpq-sz %swpocc
01:00:00      1.0      7      0.0      0
02:00:00      1.0      7      0.0      0
03:00:00      1.0      7      0.0      0
04:00:00      1.0      7      0.0      0
05:00:00      1.0      6      0.0      0
06:00:00      1.0      7      0.0      0

Average      1.0      7      0.0      0
```

### ▼ 사용되지 않은 메모리를 확인하는 방법(sar -r)

- sar -r 명령을 사용하여 현재 사용되지 않은 메모리 페이지 및 스왑 파일 디스크 블록 수를 보고합니다.

```
$ sar -r
00:00:00 freemem freeswap
01:00:00      2135      401922
```

다음 목록은 -r 옵션의 출력을 설명합니다.

freemem      명령으로 샘플링된 간격 동안 사용자 프로세스에서 사용할 수 있는 평균 메모리 페이지 수입니다. 페이지 크기는 시스템에 따라 다릅니다.

freeswap     페이지 스와핑에 사용할 수 있는 512바이트 디스크 블록 수입니다.

### 예 3-14 사용되지 않은 메모리 확인(sar -r)

다음 예는 sar -r 명령의 출력을 보여줍니다.

```
$ sar -r
SunOS balmy 5.10 Generic_144500-10 sun4v    ...

00:00:04 freemem freeswap
01:00:00      44717      1715062
02:00:01      44733      1715496
03:00:00      44715      1714746
04:00:00      44751      1715403
05:00:00      44784      1714743
06:00:00      44794      1715186
07:00:00      44793      1715159
08:00:00      44786      1714914
```

```

08:20:00  44805  1715576
08:40:01  44797  1715347
09:00:00  44761  1713948
09:20:00  44802  1715478
09:40:00  41770  1682239
10:00:00  35401  1610833
10:20:00  34295  1599141
10:40:00  33943  1598425
11:00:00  30500  1561959

Average   43312  1699242
    
```

## ▼ CPU 사용률을 확인하는 방법(sar -u)

- sar -u 명령을 사용하여 CPU 사용률 통계를 표시합니다.

```

$ sar -u
00:00:00  %usr  %sys  %wio  %idle
01:00:00      0      0      0    100
    
```

옵션이 없는 sar 명령은 sar -u 명령과 같습니다. 특정 시점에 프로세서는 사용 상태이거나 유휴 상태입니다. 사용 상태일 때 프로세서는 사용자 모드이거나 시스템 모드입니다. 유휴 상태일 때 프로세서는 I/O 완료를 대기 중이거나 할 일이 없어 “단순 유휴 상태”입니다.

다음 목록은 -u 옵션의 출력을 설명합니다.

%usr     프로세서가 사용자 모드인 시간의 백분율을 나열합니다.

%sys     프로세서가 시스템 모드인 시간의 백분율을 나열합니다.

%wio     프로세서가 유휴 상태이고 I/O 완료를 대기 중인 시간의 백분율을 나열합니다.

%idle    프로세서가 유휴 상태이고 I/O를 대기 중이 아닌 시간의 백분율을 나열합니다.

높은 %wio 값은 일반적으로 디스크 성능 저하가 발생함을 의미합니다.

### 예 3-15 CPU 사용률 확인(sar -u)

다음 예는 sar -u 명령의 출력을 나타냅니다.

```

$ sar -u
00:00:04  %usr  %sys  %wio  %idle
01:00:00      0      0      0    100
02:00:01      0      0      0    100
03:00:00      0      0      0    100
04:00:00      0      0      0    100
05:00:00      0      0      0    100
06:00:00      0      0      0    100
07:00:00      0      0      0    100
08:00:00      0      0      0    100
08:20:00      0      0      0     99
08:40:01      0      0      0     99
    
```

09:00:00	0	0	0	99
09:20:00	0	0	0	99
09:40:00	4	1	0	95
10:00:00	4	2	0	94
10:20:00	1	1	0	98
10:40:00	18	3	0	79
11:00:00	25	3	0	72
Average	2	0	0	98

### ▼ 시스템 테이블 상태를 확인하는 방법(sar -v)

- sar -v 명령을 사용하여 프로세스 테이블, inode 테이블, 파일 테이블 및 공유 메모리 레코드 테이블의 상태를 보고합니다.

```
$ sar -v
00:00:00 proc-sz   ov  inod-sz   ov  file-sz   ov  lock-sz
01:00:00  43/922     0 2984/4236  0  322/322   0  0/0
```

#### 예 3-16 시스템 테이블 상태 확인(sar -v)

다음 약식 예는 sar -v 명령의 출력을 보여줍니다. 이 예는 모든 테이블이 오버플로우가 없을 만큼 큰 경우입니다. 이러한 테이블은 물리적 메모리의 양을 기준으로 모두 동적으로 할당됩니다.

```
$ sar -v
00:00:04 proc-sz   ov  inod-sz   ov  file-sz   ov  lock-sz
01:00:00  69/8010   0 3476/34703  0  0/0       0  0/0
02:00:01  69/8010   0 3476/34703  0  0/0       0  0/0
03:00:00  69/8010   0 3476/34703  0  0/0       0  0/0
04:00:00  69/8010   0 3494/34703  0  0/0       0  0/0
05:00:00  69/8010   0 3494/34703  0  0/0       0  0/0
06:00:00  69/8010   0 3494/34703  0  0/0       0  0/0
07:00:00  69/8010   0 3494/34703  0  0/0       0  0/0
08:00:00  69/8010   0 3494/34703  0  0/0       0  0/0
08:20:00  69/8010   0 3494/34703  0  0/0       0  0/0
08:40:01  69/8010   0 3494/34703  0  0/0       0  0/0
09:00:00  69/8010   0 3494/34703  0  0/0       0  0/0
09:20:00  69/8010   0 3494/34703  0  0/0       0  0/0
09:40:00  74/8010   0 3494/34703  0  0/0       0  0/0
10:00:00  75/8010   0 4918/34703  0  0/0       0  0/0
10:20:00  72/8010   0 4918/34703  0  0/0       0  0/0
10:40:00  71/8010   0 5018/34703  0  0/0       0  0/0
11:00:00  77/8010   0 5018/34703  0  0/0       0  0/0
```

-v 옵션의 출력은 다음 표에 설명되어 있습니다.

필드 이름	설명
proc-sz	커널에서 현재 사용되고 있거나 할당된 프로세스 항목(proc 구조) 수입니다.
inod-sz	커널에서 할당된 최대 inode 수와 비교한 메모리의 총 inode 수입니다. 이 수는 엄격한 고수위가 아닙니다. 수는 오버플로우될 수 있습니다.
file-sz	열린 시스템 파일 테이블의 크기입니다. 공간이 파일 테이블에 대해 동적으로 할당되므로 sz는 0으로 지정됩니다.
ov	각 테이블에 대한 샘플링 시점 사이에 발생하는 오버플로우입니다.
lock-sz	커널에서 현재 사용되고 있거나 할당된 공유 메모리 레코드 테이블 항목 수입니다. 공간이 공유 메모리 레코드 테이블에 대해 동적으로 할당되므로 sz는 0으로 지정됩니다.

### ▼ 스와핑 작업을 확인하는 방법(sar -w)

- sar -w 명령을 사용하여 스와핑 및 전환 작업을 보고합니다.

```
$ sar -w
00:00:00 swpin/s bswin/s swpot/s bswot/s pswch/s
01:00:00 0.00 0.0 0.00 0.0 22
```

다음 목록은 sar -w 명령 출력과 관련된 목표 값 및 관찰을 설명합니다.

- swpin/s 초당 메모리로 LWP 전송 수입니다.
- bswin/s 초당 스왑 인을 위해 전송된 블록 수입니다. /\* (float)PGTOBLK(xx->cvmi.pgswapin) / sec\_diff \*/.
- swpot/s 초당 메모리에서 스왑 아웃된 평균 프로세스 수입니다. 이 수가 1보다 클 경우 메모리를 늘려야 합니다.
- bswot/s 초당 스왑 아웃을 위해 전송된 블록 수입니다.
- pswch/s 초당 커널 스레드 전환 수입니다.

주 - 모든 프로세스 스왑 인에는 프로세스 초기화가 포함됩니다.

### 예 3-17 스왑 작업 확인(sar -w)

다음 예는 sar -w 명령의 출력을 보여줍니다.

```
$ sar -w
00:00:04 swpin/s bswin/s swpot/s bswot/s pswch/s
01:00:00 0.00 0.0 0.00 0.0 132
02:00:01 0.00 0.0 0.00 0.0 133
03:00:00 0.00 0.0 0.00 0.0 133
04:00:00 0.00 0.0 0.00 0.0 134
05:00:00 0.00 0.0 0.00 0.0 133
06:00:00 0.00 0.0 0.00 0.0 133
07:00:00 0.00 0.0 0.00 0.0 132
08:00:00 0.00 0.0 0.00 0.0 131
08:20:00 0.00 0.0 0.00 0.0 133
08:40:01 0.00 0.0 0.00 0.0 132
09:00:00 0.00 0.0 0.00 0.0 132
09:20:00 0.00 0.0 0.00 0.0 132
09:40:00 0.00 0.0 0.00 0.0 335
10:00:00 0.00 0.0 0.00 0.0 601
10:20:00 0.00 0.0 0.00 0.0 353
10:40:00 0.00 0.0 0.00 0.0 747
11:00:00 0.00 0.0 0.00 0.0 804

Average 0.00 0.0 0.00 0.0 198
```

### ▼ 터미널 작업을 확인하는 방법(sar -y)

- sar -y 명령을 사용하여 터미널 장치 작업을 모니터링합니다.

```
$ sar -y
00:00:00 rawch/s canch/s outch/s rcvin/s xmtin/s mdmin/s
01:00:00 0 0 0 0 0 0
```

많은 터미널 I/O가 있을 경우 이 보고서를 사용하여 잘못된 라인이 존재하는지 확인할 수 있습니다. 기록된 작업은 다음 목록에서 정의됩니다.

rawch/s 초당 입력 문자(원시 대기열)입니다.

canch/s 초당 캐논으로 처리된 입력 문자(캐노니컬 대기열)입니다.

outch/s 초당 출력 문자(출력 대기열)입니다.

rcvin/s 초당 수신기 하드웨어 인터럽트입니다.

xmtin/s 초당 전송기 하드웨어 인터럽트입니다.

mdmin/s 초당 모뎀 인터럽트입니다.

초당 모뎀 인터럽트 수(mdmin/s)는 0에 가까워야 합니다. 초당 수신 및 전송 인터럽트(xmtin/s 및 rcvin/s)는 각각 수신 또는 전송 문자의 수보다 작거나 같아야 합니다. 그렇지 않은 경우 잘못된 라인을 확인합니다.

### 예 3-18 터미널 작업 확인(sar -y)

다음 예는 sar -y 명령의 출력을 보여줍니다.

```

$ sar -y

00:00:04 rawch/s  canch/s  outch/s  rcvin/s  xmtin/s  mdmin/s
01:00:00      0       0       0       0       0       0
02:00:01      0       0       0       0       0       0
03:00:00      0       0       0       0       0       0
04:00:00      0       0       0       0       0       0
05:00:00      0       0       0       0       0       0
06:00:00      0       0       0       0       0       0
07:00:00      0       0       0       0       0       0
08:00:00      0       0       0       0       0       0
08:20:00      0       0       0       0       0       0
08:40:01      0       0       0       0       0       0
09:00:00      0       0       0       0       0       0
09:20:00      0       0       0       0       0       0
09:40:00      0       0       1       0       0       0
10:00:00      0       0      37       0       0       0
10:20:00      0       0       0       0       0       0
10:40:00      0       0       3       0       0       0
11:00:00      0       0       3       0       0       0

Average      0       0       1       0       0       0
    
```

### ▼ 전체 시스템 성능을 확인하는 방법(sar -A)

- **sar -A** 명령을 사용하여 모든 옵션의 통계를 표시하여 전체 시스템 성능 보기를 제공합니다.

이 명령은 좀더 포괄적인 관점을 제공합니다. 단일 시점 세그먼트의 데이터보다 많은 데이터가 표시될 경우 보고서에는 평균이 포함됩니다.

## 자동으로 시스템 작업 데이터 수집(sar)

세 명령은 `sadc`, `sa1` 및 `sa2`의 시스템 작업 데이터 자동 수집에 포함됩니다.

`sadc` 데이터 수집 유틸리티는 시스템 작업에 대한 데이터를 정기적으로 수집하고 이진 형식의 파일(24시간 기간마다 하나씩)에 데이터를 저장합니다. `sadc` 명령이 정기적으로(대개 매시간마다 한번), 그리고 시스템이 다중 사용자 모드로 부트될 때마다 실행되도록 설정할 수 있습니다. 데이터 파일은 `/var/adm/sa` 디렉토리에 보관됩니다. 각 파일의 이름은 `sadd`로 지정됩니다. 여기서 `dd`는 현재 날짜입니다. 명령의 형식은 다음과 같습니다.

```
/usr/lib/sa/sadc [t n] [ofile]
```

명령은 `t`초 간격으로 `n`회 샘플링하며, 샘플링 간격은 5초보다 길어야 합니다. 그런 다음 이 명령은 이진 `ofile` 파일이나 표준 출력에 씁니다.

## 부트할 때 `sadc` 명령 실행

카운터가 0으로 재설정될 때부터 통계를 기록하려면 `sadc` 명령이 시스템 부트 시 실행되어야 합니다. `sadc` 명령이 부트 시 실행되도록 하기 위해 `svcadm enable system/sar:default` 명령이 레코드를 일별 데이터 파일에 씁니다.

명령 항목의 형식은 다음과 같습니다.

```
/usr/bin/su sys -c "/usr/lib/sa/sadc /var/adm/sa/sa`date +%d`"
```

## `sa1` 스크립트를 사용하여 정기적으로 `sadc` 명령 실행

정기적인 레코드를 생성하려면 `sadc` 명령을 정기적으로 실행해야 합니다. 이를 위한 가장 간단한 방법은 `/var/spool/cron/crontabs/sys` 파일에서 다음 줄을 주석 처리하는 것입니다.

```
# 0 * * * 0-6 /usr/lib/sa/sa1
# 20,40 8-17 * * 1-5 /usr/lib/sa/sa1
# 5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 1200 -A
```

`sys` crontab 항목은 다음을 수행합니다.

- 처음 두 crontab 항목은 월요일부터 금요일까지 오전 8시부터 오후 5시까지 20분마다(그렇지 않은 경우 매시간마다) 레코드가 `/var/adm/sa/sadd` 파일에 쓰여지도록 합니다.
- 세번째 항목은 월요일부터 금요일까지 매시간마다 레코드를 `/var/adm/sa/sar dd` 파일에 쓰고, 모든 `sar` 옵션을 포함합니다.

이러한 기본값은 필요에 맞게 변경할 수 있습니다.

## `sa2` 셸 스크립트를 사용하여 보고서 생성

또 하나의 셸 스크립트인 `sa2`는 이진 데이터 파일 대신 보고서를 생성합니다. `sa2` 명령은 `sar` 명령을 호출하고 ASCII 출력을 보고서 파일에 씁니다.

## 자동 데이터 수집 설정(`sar`)

`sar` 명령은 시스템 작업 데이터 자체를 수집하거나 `sadc` 명령으로 생성된 일별 작업 파일에서 수집된 정보를 보고하는 데 사용할 수 있습니다.

`sar` 명령의 형식은 다음과 같습니다.

```
sar [-aAbcdgkmpqruvwy] [-o file] t [n]
```

```
sar [-aAbcdgkmpqruvwy] [-s time] [-e time] [-i sec] [-f file]
```

다음 `sar` 명령은 운영 체제에서 누적 작업 카운터를  $t$  초마다  $n$  회 샘플링합니다.  $t$  는 5 초 이상이어야 합니다. 그렇지 않으면 명령 자체가 샘플에 영향을 줄 수 있습니다. 샘플링을 수행할 시간 간격을 지정해야 합니다. 그렇지 않으면 명령은 두번째 형식에 따라 작동합니다.  $n$  의 기본값은 1입니다. 다음 예는 10 초로 구분된 두 번의 샘플링을 수행합니다. `-o` 옵션이 지정된 경우 샘플이 이진 형식으로 지정됩니다.

**\$ sar -u 10 2**

`sar` 명령에 대한 기타 중요 정보에는 다음이 포함됩니다.

- 샘플링 간격이나 샘플 수가 지정되지 않으면 `sar` 명령은 이전에 기록된 파일에서 데이터를 추출합니다. 이 파일은 `-f` 옵션으로 지정된 파일이거나 기본적으로 최근의 표준 일별 작업 파일인 `/var/adm/sa/sa dd`입니다.
- `-s` 및 `-e` 옵션은 보고서에 대한 시작 시간과 종료 시간을 정의합니다. 시작 시간과 종료 시간의 형식은 `hh[:mm[:ss]]`입니다. 여기서 `hh`, `mm` 및 `ss` 는 시, 분, 초를 나타냅니다.
- `-i` 옵션은 레코드 선택 사이의 간격(초)을 지정합니다. `-i` 옵션이 포함되지 않으면 일별 작업 파일에서 발견된 모든 간격이 보고됩니다.

다음 표는 `sar` 옵션 및 해당 작업을 나열합니다.

표 3-5 sar 명령에 대한 옵션

옵션	작업
<code>-a</code>	파일 액세스 작업을 확인합니다.
<code>-b</code>	버퍼 작업을 확인합니다.
<code>-c</code>	시스템 호출을 확인합니다.
<code>-d</code>	각 블록 장치에 대한 작업을 확인합니다.
<code>-g</code>	페이지 아웃 및 메모리 해제를 확인합니다.
<code>-k</code>	커널 메모리 할당을 확인합니다.
<code>-m</code>	프로세스간 통신을 확인합니다.
<code>-nv</code>	시스템 테이블 상태를 확인합니다.
<code>-p</code>	스왑 및 전달 작업을 확인합니다.
<code>-q</code>	대기열 작업을 확인합니다.
<code>-r</code>	사용되지 않은 메모리를 확인합니다.
<code>-u</code>	CPU 사용률을 확인합니다.
<code>-w</code>	스와핑 및 전환 볼륨을 확인합니다.

표 3-5 sar 명령에 대한 옵션 (계속)

옵션	작업
-y	터미널 작업을 확인합니다.
-A	전체 시스템 성능을 보고합니다. 모든 옵션을 입력하는 것과 같습니다.

옵션을 사용하지 않는 것은 sar 명령을 -u 옵션과 함께 사용하는 것과 같습니다.

## ▼ 자동 데이터 수집을 설정하는 방법

### 1 root 역할을 맡습니다.

**Oracle Solaris 11.1 관리: 보안 서비스의 “지정된 관리 권한을 사용하는 방법”**을 참조하십시오.

### 2 svcadm enable system/sar:default 명령을 실행합니다.

이 버전의 sadc 명령은 카운터가 0으로 재설정될 때(부트 시) 시간을 표시하는 특수 레코드를 씁니다.

### 3 /var/spool/cron/crontabs/sys crontab 파일을 편집합니다.

주 - crontab 파일을 직접 편집하지 마십시오. 대신 crontab -e 명령을 사용하여 기존 crontab 파일을 변경합니다.

```
# crontab -e sys
```

### 4 다음 줄을 주석 처리합니다.

```
0 * * * 0-6 /usr/lib/sa/sa1
20,40 8-17 * * 1-5 /usr/lib/sa/sa1
5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 1200 -A
```

자세한 내용은 [crontab\(1\)](#) 매뉴얼 페이지를 참조하십시오.



## 시스템 작업 예약(작업)

---

이 장에서는 `crontab` 및 `at` 명령을 사용하여 루틴 또는 단일(일회성) 시스템 작업을 예약하는 방법을 설명합니다.

또한 다음 파일을 사용하여 이러한 명령에 대한 액세스를 제어하는 방법을 설명합니다.

- `cron.deny`
- `cron-allow`
- `at.deny`

다음은 이 장에 포함된 정보 목록입니다.

- 81 페이지 “시스템 작업을 자동으로 실행하는 방법”
- 83 페이지 “시스템 작업 예약”
- 93 페이지 “`at` 명령을 사용하여 작업 예약”

## 시스템 작업을 자동으로 실행하는 방법

많은 시스템 작업을 자동으로 실행하도록 설정할 수 있습니다. 이러한 작업 중 일부는 정기적 간격으로 발생해야 합니다. 다른 작업은 한 번만, 보통 저녁 또는 주말에 근무 외 시간 중 실행해야 합니다.

이 단원에서는 `crontab` 및 `at`의 두 명령에 대한 개요 정보를 다룹니다. 이러한 명령으로 루틴 작업이 자동으로 실행되도록 예약할 수 있습니다. `crontab` 명령은 반복적 명령을 예약합니다. `at` 명령은 한 번 실행하는 작업을 예약합니다.

다음 표는 `crontab` 및 `at` 명령과 이러한 명령에 대한 액세스를 제어하는 파일을 요약합니다.

표 4-1 명령 요약: 시스템 작업 예약

명령	예약대상	파일 위치	액세스 제어 파일
crontab	정기적 간격으로 다중 시스템 작업	/var/spool/cron/crontabs	/etc/cron.d/cron.allow 및 /etc/cron.d/cron.deny
at	단일 시스템 작업	/var/spool/cron/atjobs	/etc/cron.d/at.deny

## 반복적 작업 예약: crontab

crontab 명령을 사용하여 루틴 시스템 관리 작업이 매일, 매주 또는 매월 실행되도록 예약할 수 있습니다.

일별 crontab 시스템 관리 작업은 다음을 포함할 수 있습니다.

- 임시 디렉토리에서 며칠 이상 지난 파일 제거
- 계산 요약 명령 실행
- df 및 ps 명령을 사용하여 시스템 스냅샷 생성
- 일별 보안 모니터링 수행
- 시스템 백업 실행

주별 crontab 시스템 관리 작업은 다음을 포함할 수 있습니다.

- man -k 명령에서 사용할 catman 데이터베이스 재구성
- fsck -n 명령을 실행하여 디스크 문제 나열

월별 crontab 시스템 관리 작업은 다음을 포함할 수 있습니다.

- 특정 월 동안 사용되지 않은 파일 나열
- 월별 계산 보고서 생성

추가로 사용자는 미리 알림 보내기 및 백업 파일 제거와 같은 다른 루틴 시스템 작업을 실행하도록 crontab 명령을 예약할 수 있습니다.

crontab 작업 예약에 대한 단계별 지침은 87 페이지 “crontab 파일을 만들거나 편집하는 방법”을 참조하십시오.

## 단일 작업 예약: at

at 명령으로 나중에 실행할 작업을 예약할 수 있습니다. 작업은 단일 명령 또는 스크립트로 구성할 수 있습니다.

crontab과 마찬가지로, at 명령으로 루틴 작업의 자동 실행을 예약할 수 있습니다. 그러나 crontab 파일과 달리, at 파일은 작업을 한 번 실행한 후에 디렉토리에서 제거됩니다. 따라서 at 명령은 나중에 조사를 위해 별도의 파일로 출력을 재지정하는 단순 명령 또는 스크립트를 실행하는 데 가장 유용합니다.

at 작업을 제출하려면 명령을 입력하고 at 명령 구문에 따라 작업을 실행할 시간을 예약하는 옵션을 지정하면 됩니다. at 작업 제출에 대한 자세한 내용은 94 페이지 “at 명령의 설명”을 참조하십시오.

at 명령은 실행된 명령 또는 스크립트를 현재 환경 변수의 복사본과 함께 /var/spool/cron/atjobs 디렉토리에 저장합니다. at 작업 파일 이름은 at 대기열의 위치를 지정하는 긴 숫자 뒤에 .a 확장자를 붙여서 만듭니다(예: 793962000.a).

cron 데몬은 시작 시 at 작업을 검사하고 제출된 새 작업을 수신합니다. cron 데몬이 at 작업을 실행한 후에 at 작업의 파일이 atjobs 디렉토리에서 제거됩니다. 자세한 내용은 at(1) 매뉴얼 페이지를 참조하십시오.

at 작업 예약에 대한 단계별 지침은 95 페이지 “at 작업을 만드는 방법”을 참조하십시오.

## 시스템 작업 예약

이 절에는 crontab 파일을 사용하여 시스템 작업을 예약하기 위한 작업이 포함됩니다.

### crontab 파일 만들기 및 편집(작업 맵)

작업	설명	수행 방법
crontab 파일 만들기 또는 편집	crontab -e 명령을 사용하여 crontab 파일을 만들거나 편집합니다.	87 페이지 “crontab 파일을 만들거나 편집하는 방법”
crontab 파일이 존재하는지 확인	ls -l 명령을 사용하여 /var/spool/cron/crontabs 파일의 내용을 확인합니다.	88 페이지 “crontab 파일이 존재하는지 확인하는 방법”
crontab 파일 표시	ls -l 명령을 사용하여 crontab 파일을 표시합니다.	88 페이지 “crontab 파일을 표시하는 방법”
crontab 파일 제거	crontab 파일은 제한적 권한으로 설정됩니다. rm 명령이 아닌 crontab -r 명령을 사용하여 crontab 파일을 제거합니다.	89 페이지 “crontab 파일을 제거하는 방법”
crontab 액세스 거부	crontab 명령에 대한 사용자 액세스를 거부하려면 /etc/cron.d/cron.deny 파일을 편집하여 이 파일에 사용자 이름을 추가합니다.	91 페이지 “crontab 명령 액세스를 거부하는 방법”

작업	설명	수행 방법
crontab 액세스를 지정된 사용자로 제한	crontab 명령에 대한 사용자 액세스를 허용하려면 /etc/cron.d/cron.allow 파일에 사용자 이름을 추가합니다.	92 페이지 “ <a href="#">crontab 명령 액세스를 지정된 사용자로 제한하는 방법</a> ”

## 반복적 시스템 작업 예약(cron)

다음 단원에서는 crontab 파일을 만들고, 편집, 표시, 제거하는 방법과 이들의 액세스를 제어하는 방법을 설명합니다.

### crontab 파일 내부

cron 데몬은 각 crontab 파일에서 발견된 명령에 따라 시스템 작업을 예약합니다. crontab 파일은 정기적 간격으로 실행될 명령들(한 라인에 명령 하나씩)로 구성됩니다. 각 라인의 시작 부분은 명령을 실행할 때 cron 데몬을 알려주는 날짜 및 시간 정보를 포함합니다.

예를 들어, root라는 crontab 파일이 SunOS 소프트웨어 설치 중 제공됩니다. 파일 내용에 다음 명령줄이 포함됩니다.

```
10 3 * * * /usr/sbin/logadm      (1)
15 3 * * 0 /usr/lib/fs/nfs/nfsfind  (2)
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1  (3)
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean  (4)
```

다음은 이러한 명령줄의 출력을 설명합니다.

- 첫번째 라인은 매일 오전 3:10에 logadm 명령을 실행합니다.
- 두번째 라인은 매주 일요일 오전 3:15에 nfsfind 스크립트를 실행합니다.
- 세번째 라인은 매일 오전 2:10에 일광 절약 시간을 검사하는(필요한 경우 수정하는) 스크립트를 실행합니다.

RTC 시간대나 /etc/rtc\_config 파일이 없는 경우 이 항목은 아무것도 아닙니다.

---

**x86 전용** - /usr/sbin/rtc 스크립트는 x86 기반 시스템에서만 실행할 수 있습니다.

---

- 네번째 라인은 매일 오전 3:30에 일반 보안 서비스 테이블 /etc/gss/gsscred\_db에서 중복 항목을 검사하고 제거합니다.

crontab 파일 내의 라인 구문에 대한 자세한 내용은 85 페이지 “[crontab 파일 항목의 구문](#)”을 참조하십시오.

crontab 파일은 /var/spool/cron/crontabs 디렉토리에 저장됩니다. root 외에 여러 crontab 파일이 SunOS 소프트웨어 설치 중 제공됩니다. 다음 표를 참조하십시오.

표 4-2 기본 crontab 파일

crontab 파일	기능
adm	계산
root	일반 시스템 함수 및 파일 시스템 정리
sys	성능 데이터 수집
uucp	일반 uucp 정리

기본 crontab 파일 외에, 사용자는 crontab 파일을 만들어 자신의 시스템 작업을 예약할 수 있습니다. 다른 crontab 파일은 bob, mary, smith, jones와 같은 생성된 사용자 계정에서 이름을 따옵니다.

root 또는 다른 사용자에게 속하는 crontab 파일에 액세스하려면 슈퍼유저 권한이 필요합니다.

crontab 파일을 만들고, 편집, 표시, 제거하는 방법을 설명하는 절차는 후속 단원에서 설명합니다.

## cron 데몬이 예약을 처리하는 방법

cron 데몬은 crontab 명령의 자동 예약을 관리합니다. cron 데몬의 역할은 `/var/spool/cron/crontab` 디렉토리에 crontab 파일이 있는지 검사하는 것입니다.

cron 데몬은 시작 시 다음 작업을 수행합니다.

- 새 crontab 파일을 검사합니다.
- 파일 내에 나열된 실행 시간을 읽습니다.
- 적절한 시간에 실행할 명령을 제출합니다.
- 업데이트된 crontab 파일에 관해 crontab 명령에서 알림을 받습니다.

대부분 똑같은 방법으로 cron 데몬은 at 파일의 예약을 제어합니다. 이러한 파일은 `/var/spool/cron/atjobs` 디렉토리에 저장됩니다. 또한 cron 데몬은 제출된 at 작업에 관해 crontab 명령에서 알림을 받습니다.

## crontab 파일 항목의 구문

crontab 파일은 각 명령줄의 처음 5개 필드로 지정된 시간에 자동으로 실행하는 명령들(한 라인에 명령 하나씩)로 구성됩니다. 다음 표에 설명된 이러한 5개 필드는 공백으로 구분됩니다.

표 4-3 crontab 시간 필드에 허용 가능한 값

시간 필드	값
분	0-59

표 4-3 crontab 시간 필드에 허용 가능한 값 (계속)

시간 필드	값
시	0-23
월의 일	1-31
월	1-12
주의 일	0-6 (0 = 일요일)

crontab 시간 필드에서 특수 문자를 사용할 때 다음 지침을 따르십시오.

- 공백을 사용하여 각 필드를 구분합니다.
- 逗를 사용하여 여러 값을 구분합니다.
- 하이픈을 사용하여 값 범위를 지정합니다.
- 별표를 와일드카드로 사용하여 모든 가능한 값을 포함합니다.
- 라인 시작 부분에 주석 마크(#)를 사용하여 주석이나 빈 라인을 나타냅니다.

예를 들어, 다음 crontab 명령 입력은 매월 1일과 15일, 오후 4시에 사용자 콘솔 창에 미리 알림을 표시합니다.

```
0 16 1,15 * * echo Timesheets Due > /dev/console
```

crontab 파일의 각 명령은 너무 길더라도 한 라인씩 구성해야 합니다. crontab 파일은 여분의 캐리지 리턴을 인식하지 않습니다. crontab 항목 및 명령 옵션에 대한 자세한 내용은 [crontab\(1\)](#) 매뉴얼 페이지를 참조하십시오.

## crontab 파일 만들기 및 편집

crontab 파일을 만드는 가장 간단한 방법은 `crontab -e` 명령을 사용하는 것입니다. 이 명령은 시스템 환경에 설정된 텍스트 편집기를 호출합니다. 시스템 환경의 기본 편집기는 `EDITOR` 환경 변수에 정의됩니다. 이 변수가 설정되지 않은 경우 `crontab` 명령이 기본 편집기인 `ed`를 사용합니다. 가급적 잘 아는 편집기를 선택해야 합니다.

다음 예는 편집기가 정의되었는지 확인하는 방법과 `vi`를 기본값으로 설정하는 방법을 보여줍니다.

```
$ which $EDITOR
$
$ EDITOR=vi
$ export EDITOR
```

crontab 파일을 만들 때 `/var/spool/cron/crontabs` 디렉토리에 자동으로 놓이고 사용자 이름이 부여됩니다. 루트 권한이 있는 경우 다른 사용자 또는 `root`에 대해 crontab 파일을 만들거나 편집할 수 있습니다.

## ▼ crontab 파일을 만들거나 편집하는 방법

시작하기 전에 root 또는 다른 사용자에게 속하는 crontab 파일을 만들거나 편집하는 경우 root 역할을 가져야 합니다. [Oracle Solaris 11.1 관리: 보안 서비스의 “지정된 관리 권한을 사용하는 방법”](#)을 참조하십시오.

자신의 고유 crontab 파일을 편집할 때는 root 역할을 가질 필요가 없습니다.

### 1 새 crontab 파일을 만들거나 기존 파일을 편집합니다.

```
# crontab -e [username]
```

여기서 *username*은 crontab 파일을 만들거나 편집할 사용자 계정의 이름을 지정합니다. 슈퍼유저 권한 없이 자신의 crontab 파일을 만들 수 있지만, root 또는 다른 사용자의 crontab 파일을 만들거나 편집하려면 슈퍼유저 권한이 있어야 합니다.



주의 - 실수로 crontab 명령을 아무 옵션 없이 입력한 경우 편집기의 인터럽트 문자를 누르십시오. 이 문자는 변경 사항을 저장하지 않고 종료하도록 허용합니다. 대신, 변경 사항을 저장하고 파일을 종료한 경우 기존 crontab 파일이 빈 파일로 겹쳐 쓰입니다.

### 2 crontab 파일에 명령줄을 추가합니다.

85 페이지 “[crontab 파일 항목의 구문](#)”에 설명된 구문을 따릅니다. crontab 파일은 `/var/spool/cron/crontabs` 디렉토리에 놓입니다.

### 3 crontab 파일 변경 사항을 확인합니다.

```
# crontab -l [username]
```

## 예 4-1 crontab 파일 만들기

다음 예는 다른 사용자의 crontab 파일을 만드는 방법을 보여줍니다.

```
# crontab -e jones
```

새 crontab 파일에 자동으로 추가된 다음 명령 입력은 매주 일요일 아침, 오전 1:00에 사용자의 홈 디렉토리에서 로그 파일을 제거합니다. 명령 입력이 출력을 재지정하지 않기 때문에 \*.log 뒤의 명령줄에 재지정 문자가 추가됩니다. 이렇게 하면 명령이 올바르게 실행됩니다.

```
# This command helps clean up user accounts.
1 0 * * 0 rm /home/jones/*.log > /dev/null 2>&1
```

## ▼ crontab 파일이 존재하는지 확인하는 방법

- 사용자에 대해 crontab 파일이 존재하는지 확인하려면 /var/spool/cron/crontabs 디렉토리에 ls -l 명령을 사용합니다. 예를 들어, 다음 출력은 사용자 jones 및 smith에 대해 crontab 파일이 존재함을 보여줍니다.

```
$ ls -l /var/spool/cron/crontabs
```

88 페이지 “crontab 파일을 표시하는 방법”에 설명된 crontab -l 명령을 사용하여 사용자의 crontab 파일 내용을 확인합니다.

## crontab 파일 표시

crontab -l 명령은 cat 명령이 다른 유형의 파일 내용을 표시하는 것과 똑같은 방법으로 crontab 파일의 내용을 표시합니다. 이 명령을 사용하기 위해 디렉토리를 /var/spool/cron/crontabs(여기에 crontab 파일이 위치함)로 변경할 필요는 없습니다.

기본적으로 crontab -l 명령은 자신의 crontab 파일을 표시합니다. 다른 사용자에 속하는 crontab 파일을 표시하려면 슈퍼유저여야 합니다.

## ▼ crontab 파일을 표시하는 방법

시작하기 전에 root 또는 다른 사용자에게 속하는 crontab 파일을 표시하려면 root 역할을 맡습니다. **Oracle Solaris 11.1 관리: 보안 서비스의 “지정된 관리 권한을 사용하는 방법”**을 참조하십시오.

자신의 고유 crontab 파일을 표시할 때는 root 역할을 가질 필요가 없습니다.

- crontab 파일을 표시합니다.

```
# crontab -l [username]
```

여기서 *username*은 crontab 파일을 표시할 사용자 계정의 이름을 지정합니다. 다른 사용자의 crontab 파일을 표시하려면 슈퍼유저 권한이 필요합니다.



**주의** - 실수로 crontab 명령을 아무 옵션 없이 입력한 경우 편집기의 인터럽트 문자를 누르십시오. 이 문자는 변경 사항을 저장하지 않고 종료하도록 허용합니다. 대신, 변경 사항을 저장하고 파일을 종료한 경우 기존 crontab 파일이 빈 파일로 겹쳐 쓰입니다.

### 예 4-2 crontab 파일 표시

이 예는 crontab -l 명령을 사용하여 사용자의 기본 crontab 파일의 내용을 표시하는 방법을 보여줍니다.

```
$ crontab -l
13 13 * * * chmod g+w /home1/documents/*.book > /dev/null 2>&1
```

### 예 4-3 기본 root crontab 파일 표시

이 예는 기본 root crontab 파일을 표시하는 방법을 보여줍니다.

```
$ suPassword:
# crontab -l
#ident "@(#)root      1.19   98/07/06 SMI"   /* SVr4.0 1.1.3.1   */
#
# The root crontab should be used to perform accounting data collection.
#
#
10 3 * * * /usr/sbin/logadm
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
#10 3 * * * /usr/lib/krb5/kprop_script ___slave_kdcs___
```

### 예 4-4 다른 사용자의 crontab 파일 표시

이 예는 다른 사용자에게 속하는 crontab 파일을 표시하는 방법을 보여줍니다.

```
$ su
Password:
# crontab -l jones
13 13 * * * cp /home/jones/work_files /usr/backup/. > /dev/null 2>&1
```

## crontab 파일 제거

기본적으로 crontab 파일 보호가 설정되므로 rm 명령으로 crontab 파일을 부주의하게 삭제할 수 없습니다. 대신, crontab -r 명령을 사용하여 crontab 파일을 제거하십시오.

기본적으로 crontab -r 명령은 자신의 crontab 파일을 제거합니다.

이 명령을 사용하기 위해 디렉토리를 /var/spool/cron/crontabs(여기에 crontab 파일이 위치함)로 변경할 필요는 없습니다.

### ▼ crontab 파일을 제거하는 방법

시작하기 전에 root 또는 다른 사용자에게 속하는 crontab 파일을 제거하려면 root 역할을 맡습니다. 역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. [Oracle Solaris 11.1 관리: 보안 서비스의 “지정된 관리 권한을 사용하는 방법”](#)을 참조하십시오.

자신의 고유 crontab 파일을 제거할 때는 root 역할을 가질 필요가 없습니다.

#### 1 crontab 파일을 제거합니다.

```
# crontab -r [username]
```

여기서 *username*은 crontab 파일을 제거할 사용자 계정의 이름을 지정합니다. 다른 사용자의 crontab 파일을 제거하려면 슈퍼유저 권한이 필요합니다.



주의 - 실수로 `crontab` 명령을 아무 옵션 없이 입력한 경우 편집기의 인터럽트 문자를 누르십시오. 이 문자는 변경 사항을 저장하지 않고 종료하도록 허용합니다. 대신, 변경 사항을 저장하고 파일을 종료한 경우 기존 `crontab` 파일이 빈 파일로 겹쳐 쓰입니다.

## 2 crontab 파일이 제거되었는지 확인합니다.

```
# ls /var/spool/cron/crontabs
```

### 예 4-5 crontab 파일 제거

다음 예에서는 사용자 `smith`가 `crontab -r` 명령을 사용하여 자신의 `crontab` 파일을 제거하는 방법을 보여줍니다.

```
$ ls /var/spool/cron/crontabs
adm   jones   root    smith   sys     uucp
$ crontab -r
$ ls /var/spool/cron/crontabs
adm   jones   root    sys     uucp
```

## crontab 명령에 대한 액세스 제어

`/etc/cron.d` 디렉토리의 두 파일 `cron.deny` 및 `cron.allow`를 사용하여 `crontab` 명령에 대한 액세스를 제어할 수 있습니다. 이러한 파일은 지정된 사용자만 자신의 `crontab` 파일 만들기, 편집, 표시, 제거와 같은 `crontab` 명령 작업을 수행하도록 허용합니다.

`cron.deny` 및 `cron.allow` 파일은 사용자 이름 목록(한 라인에 사용자 이름 하나씩)으로 구성됩니다.

이러한 액세스 제어 파일은 다음과 같이 작동합니다.

- `cron.allow`가 있는 경우 이 파일에 나열된 사용자만 `crontab` 파일을 만들거나, 편집, 표시, 제거할 수 있습니다.
- `cron.allow`가 없는 경우 `cron.deny`에 나열된 사용자를 제외한 모든 사용자가 `crontab` 파일을 제출할 수 있습니다.
- `cron.allow`도 없고 `cron.deny`도 없는 경우 `crontab` 명령을 실행하려면 슈퍼유저 권한이 필요합니다.

`cron.deny` 및 `cron.allow` 파일을 편집하거나 만들려면 슈퍼유저 권한이 필요합니다.

SunOS 소프트웨어 설치 중 생성된 `cron.deny` 파일은 다음 사용자 이름을 포함합니다.

```
$ cat /etc/cron.d/cron.deny
daemon
bin
smtp
nuucp
```

```
listen
nobody
noaccess
```

기본 `cron.deny` 파일의 사용자 이름 중 아무도 `crontab` 명령에 액세스할 수 없습니다. 이 파일을 편집하여 `crontab` 명령에 액세스가 거부될 다른 사용자 이름을 추가할 수 있습니다.

기본 `cron.allow` 파일은 제공되지 않습니다. 그래서 Oracle Solaris 소프트웨어 설치 후에 모든 사용자(기본 `cron.deny` 파일에 나열된 사용자 제외)가 `crontab` 명령에 액세스할 수 있습니다. `cron.allow` 파일을 만들면 이러한 사용자만 `crontab` 명령에 액세스할 수 있습니다.

## ▼ crontab 명령 액세스를 거부하는 방법

### 1 root 역할을 맡습니다.

**Oracle Solaris 11.1 관리: 보안 서비스의 “지정된 관리 권한을 사용하는 방법”을 참조하십시오.**

```
$ su -
Password:
#
```

### 2 /etc/cron.d/cron.deny 파일을 편집하고 사용자 이름을 한 라인에 하나씩 추가합니다. crontab 명령에 대한 액세스가 거부될 사용자를 넣습니다.

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
username1
username2
username3
.
.
.
```

### 3 /etc/cron.d/cron.deny 파일이 새 항목을 포함하는지 확인합니다.

```
# cat /etc/cron.d/cron.deny
daemon
bin
nuucp
listen
nobody
noaccess
```

## ▼ crontab 명령 액세스를 지정된 사용자로 제한하는 방법

- 1 root 역할을 맡습니다.

**Oracle Solaris 11.1 관리: 보안 서비스의 “지정된 관리 권한을 사용하는 방법”**을 참조하십시오.

- 2 /etc/cron.d/cron.allow 파일을 만듭니다.

- 3 root 역할을 cron.allow 파일에 추가합니다.

root를 파일에 추가하지 않으면 crontab 명령에 대한 슈퍼유저 액세스가 거부됩니다.

- 4 사용자 이름을 한 라인에 하나씩 추가합니다.

crontab 명령을 사용하도록 허용될 사용자를 넣습니다.

```
root
username1
username2
username3
.
.
.
```

### 예 4-6 crontab 명령 액세스를 지정된 사용자로 제한

다음 예는 사용자 이름 jones, temp, visitor가 crontab 명령에 액세스하지 못하게 막는 cron.deny 파일을 보여줍니다.

```
$ cat /etc/cron.d/cron.deny
daemon
bin
smtp
nuucp
listen
nobody
noaccess
jones
temp
visitor
```

다음 예는 cron.allow 파일을 보여줍니다. 사용자 root, jones 및 smith는 crontab 명령에 액세스할 수 있는 유일한 사용자입니다.

```
$ cat /etc/cron.d/cron.allow
root
jones
smith
```

## 제한된 crontab 명령 액세스를 확인하는 방법

특정 사용자가 crontab 명령에 액세스할 수 있는지 확인하려면 사용자 계정으로 로그인한 동안 `crontab -l` 명령을 사용합니다.

**\$ crontab -l**

사용자가 crontab 명령에 액세스할 수 있고 이미 crontab 파일을 만든 경우 파일이 표시됩니다. 그렇지 않으면, 사용자가 crontab 명령에 액세스할 수 있지만 crontab 파일이 없어서 다음과 비슷한 메시지가 표시됩니다.

```
crontab: can't open your crontab file
```

이 사용자가 `cron.allow` 파일(존재하는 경우)에 나열되거나 사용자가 `cron.deny` 파일에 나열되지 않습니다.

사용자가 crontab 명령에 액세스할 수 없을 경우 다음 메시지가 이전 crontab 파일이 존재하는지 여부를 표시합니다.

```
crontab: you are not authorized to use cron. Sorry.
```

이 메시지는 사용자가 `cron.allow` 파일(있는 경우)에 나열되지 않거나 사용자가 `cron.deny` 파일에 나열되어 있음을 의미합니다.

## at 명령을 사용하여 작업 예약

이 절에는 at 명령을 사용하여 시스템 작업을 예약하기 위한 작업이 포함됩니다.

### at 명령 사용(작업 맵)

작업	설명	수행 방법
at 작업 만들기	<p>at 명령을 사용하여 다음을 수행합니다.</p> <ul style="list-style-type: none"> <li>명령줄에서 at 유틸리티를 시작합니다.</li> <li>실행할 명령 또는 스크립트를 한 라인에 하나씩 입력합니다.</li> <li>at 유틸리티를 종료하고 작업을 저장합니다.</li> </ul>	95 페이지 “at 작업을 만드는 방법”
at 대기열 표시	atq 명령을 사용하여 at 대기열을 표시합니다.	96 페이지 “at 대기열을 표시하는 방법”

작업	설명	수행 방법
at 작업 확인	atq 명령을 사용하여 특정 사용자에게 속하는 at 작업이 대기열에 제출되었는지 확인합니다.	96 페이지 “at 작업을 확인하는 방법”
at 작업 표시	at -l [job-id] 명령을 사용하여 대기열에 제출된 at 작업을 표시합니다.	96 페이지 “at 작업을 표시하는 방법”
at 작업 제거	at -r [job-id] 명령을 사용하여 대기열에서 at 작업을 제거합니다.	97 페이지 “at 작업을 제거하는 방법”
at 명령에 대한 액세스 거부	at 명령에 대한 사용자 액세스를 거부하려면 /etc/cron.d/at.deny 파일을 편집합니다.	98 페이지 “at 명령에 대한 액세스를 거부하는 방법”

## 단일 시스템 작업 예약(at)

다음 단원에서는 at 명령을 사용하여 다음 작업을 수행하는 방법을 설명합니다.

- 나중에 실행할 작업(명령 및 스크립트) 예약
- 이러한 작업을 표시 및 제거하는 방법
- at 명령에 대한 액세스를 제어하는 방법

기본적으로 사용자는 자신의 at 작업 파일을 만들고, 표시 및 제거할 수 있습니다. root 또는 다른 사용자에게 속하는 at 파일에 액세스하려면 슈퍼유저 권한이 있어야 합니다.

at 작업을 제출할 때 작업 식별 번호가 .a 확장자와 함께 지정됩니다. 이 지정이 작업의 파일 이름과 대기열 번호가 됩니다.

### at 명령의 설명

at 작업 파일 제출은 다음 단계를 따릅니다.

1. at 유틸리티를 호출하고 명령 실행 시간을 지정합니다.
2. 나중에 실행할 명령 또는 스크립트를 입력합니다.

---

주 - 이 명령 또는 스크립트의 출력이 중요하면 나중에 조사할 수 있도록 출력을 파일로 재지정해야 합니다.

---

예를 들어, 다음 at 작업은 7월의 마지막 날 자정에 사용자 계정 smith에서 core 파일을 제거합니다.

```
$ at 11:45pm July 31
at> rm /home/smith/*core*
at> Press Control-d
commands will be executed using /bin/csh
job 933486300.a at Tue Jul 31 23:45:00 2004
```

## at 명령에 대한 액세스 제어

지정된 사용자만 at 작업에 대한 대기열 정보를 만들거나, 제거, 표시하도록 허용하여 at 명령에 대한 액세스를 제어하도록 파일을 설정할 수 있습니다. at 명령에 대한 액세스를 제어하는 파일인 /etc/cron.d/at.deny는 사용자 이름 목록(한 라인에 사용자 이름 하나씩)으로 구성됩니다. 이 파일에 나열된 사용자는 at 명령에 액세스할 수 없습니다.

SunOS 소프트웨어 설치 중 생성된 at.deny 파일은 다음 사용자 이름을 포함합니다.

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
```

수퍼유저 권한으로 at.deny 파일을 편집하여 제한할 at 명령 액세스를 가진 다른 사용자 이름을 추가할 수 있습니다.

## ▼ at 작업을 만드는 방법

- 1 at 유틸리티를 시작하여 작업을 실행할 시간을 지정합니다.

```
$ at [-m] time [date]
```

**-m**      작업이 완료된 후에 전자 메일을 보냅니다.

**time**    작업을 예약할 시를 지정합니다. 24 시간제에 따라 시를 지정하지 않으려면 am 또는 pm을 추가합니다. 허용 가능한 키워드는 midnight, noon, now입니다. 분은 선택 사항입니다.

**date**    월의 처음 3자 이상, 주의 일 또는 키워드 today나 tomorrow를 지정합니다.

- 2 at 프롬프트에서 실행할 명령 또는 스크립트를 한 라인에 하나씩 입력합니다.

각 라인의 끝에 Return을 누르면 여러 개의 명령을 입력할 수 있습니다.

- 3 at 유틸리티를 종료하고 Ctrl-D를 눌러 at 작업을 저장합니다.

at 작업에 대기열 번호(작업의 파일 이름)가 지정됩니다. 이 번호는 at 유틸리티를 종료할 때 표시됩니다.

## 예 4-7 at 작업 만들기

다음 예는 사용자 jones가 오후 7:30에 백업 파일을 제거하기 위해 만든 at 작업을 보여줍니다. 사용자는 작업이 완료된 후에 전자 메일 메시지를 받도록 -m 옵션을 사용했습니다.

```
$ at -m 1930
at> rm /home/jones/*.backup
at> Press Control-D
job 897355800.a at Thu Jul 12 19:30:00 2004
```

at 작업 실행을 확인한 전자 메일 메시지를 받았습니다.

```
Your "at" job "rm /home/jones/*.backup"
completed.
```

다음 예는 토요일 아침, 오전 4:00에 jones가 대규모 at 작업을 예약하는 방법을 보여줍니다. 작업 출력이 big.file이라는 파일로 재지정되었습니다.

```
$ at 4 am Saturday
at> sort -r /usr/dict/words > /export/home/jones/big.file
```

### ▼ at 대기열을 표시하는 방법

- at 대기열에 대기 중인 작업을 확인하려면 atq 명령을 사용합니다.

```
$ atq
```

이 명령은 생성된 at 작업에 대한 상태 정보를 표시합니다.

### ▼ at 작업을 확인하는 방법

- at 작업을 만들었는지 확인하려면 atq 명령을 사용합니다. 다음 예에서 atq 명령은 jones에 속하는 at 작업이 대기열에 제출되었음을 확인합니다.

```
$ atq
Rank   Execution Date   Owner   Job           Queue   Job Name
1st    Jul 12, 2004 19:30  jones  897355800.a   a       stdin
2nd    Jul 14, 2004 23:45  jones  897543900.a   a       stdin
3rd    Jul 17, 2004 04:00  jones  897732000.a   a       stdin
```

### ▼ at 작업을 표시하는 방법

- at 작업의 실행 시간에 대한 정보를 표시하려면 at -l 명령을 사용합니다.

```
$ at -l [job-id]
```

여기서 -l job-id 옵션은 상태를 표시할 작업의 식별 번호를 나타냅니다.

## 예 4-8 at 작업 표시

다음 예에서는 사용자가 제출한 모든 작업의 상태 정보를 제공하는 `at -l` 명령의 출력을 보여줍니다.

```
$ at -l
897543900.a    Sat Jul 14 23:45:00 2004
897355800.a    Thu Jul 12 19:30:00 2004
897732000.a    Tue Jul 17 04:00:00 2004
```

다음 예는 `at -l` 명령으로 단일 작업을 지정할 때 표시되는 출력을 보여줍니다.

```
$ at -l 897732000.a
897732000.a    Tue Jul 17 04:00:00 2004
```

## ▼ at 작업을 제거하는 방법

시작하기 전에 root 또는 다른 사용자에게 속하는 `at` 작업을 제거하려면 root 역할을 맡습니다. 역할에는 권한 부여 및 권한이 있는 명령이 포함됩니다. [Oracle Solaris 11.1 관리: 보안 서비스의 “지정된 관리 권한을 사용하는 방법”](#)을 참조하십시오.

자신의 고유 `at` 작업을 제거할 때는 root 역할을 가질 필요가 없습니다.

### 1 작업을 실행하기 전에 대기열에서 at 작업을 제거합니다.

```
# at -r [job-id]
```

여기서 `-r job-id` 옵션은 제거할 작업의 식별 번호를 지정합니다.

### 2 at -l(또는 atq) 명령을 사용하여 at 작업이 제거되었는지 확인합니다.

`at -l` 명령은 `at` 대기열에 남은 작업을 표시합니다. 지정한 식별 번호를 가진 작업이 나타나지 않아야 합니다.

```
$ at -l [job-id]
```

## 예 4-9 at 작업 제거

다음 예에서 사용자는 7월 17일 오전 4시에 실행하도록 예약된 `at` 작업을 제거하려고 합니다. 먼저, 사용자가 작업 식별 번호를 찾기 위해 `at` 대기열을 표시합니다. 그 다음, 사용자가 `at` 대기열에서 이 작업을 제거합니다. 마지막으로, 사용자가 이 작업이 대기열에서 제거되었는지 확인합니다.

```
$ at -l
897543900.a    Sat Jul 14 23:45:00 2003
897355800.a    Thu Jul 12 19:30:00 2003
897732000.a    Tue Jul 17 04:00:00 2003
$ at -r 897732000.a
$ at -l 897732000.a
at: 858142000.a: No such file or directory
```

## ▼ at 명령에 대한 액세스를 거부하는 방법

- 1 root 역할을 맡습니다.

**Oracle Solaris 11.1 관리: 보안 서비스의 “지정된 관리 권한을 사용하는 방법”을 참조하십시오.**

- 2 `/etc/cron.d/at.deny` 파일을 편집하고 at 명령의 사용이 금지될 사용자 이름을 한 라인에 하나씩 추가합니다.

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
username1
username2
username3
.
.
.
```

### 예 4-10 at 액세스 거부

다음 예는 사용자 smith 및 jones가 at 명령에 액세스할 수 없도록 편집한 `at.deny` 파일을 보여줍니다.

```
$ cat at.deny
daemon
bin
smtp
nuucp
listen
nobody
noaccess
jones
smith
```

## ▼ at 명령 액세스가 거부되었는지 확인하는 방법

- 사용자 이름이 `/etc/cron.d/at.deny` 파일에 올바르게 추가되었는지 확인하려면 사용자로 로그인한 동안 `at -l` 명령을 사용합니다. 사용자 smith가 at 명령에 액세스할 수 없으면 다음 명령이 표시됩니다.

```
# su smith
Password:
# at -l
at: you are not authorized to use at. Sorry.
```

마찬가지로, 사용자가 at 작업을 제출하려면 다음 메시지가 표시됩니다.

```
# at 2:30pm
```

```
at: you are not authorized to use at. Sorry.
```

이 메시지는 사용자가 at.deny 파일에 나열되었음을 확인합니다.

at 명령 액세스가 허용된 경우 at -l 명령이 아무것도 반환하지 않습니다.



## 시스템 콘솔, 터미널 장치 및 전원 서비스 관리(작업)

---

이 장에서는 `ttymon` 프로그램 및 시스템 전원 서비스를 통해 시스템 콘솔 및 로컬로 연결된 터미널 장치를 관리하는 방법에 대해 설명합니다.

다음은 이 장에 포함된 정보 목록입니다.

- 101 페이지 “시스템 콘솔, 터미널 장치 및 전원 서비스를 관리하기 위한 새로운 기능”
- 102 페이지 “시스템 콘솔 및 로컬로 연결된 터미널 장치 관리”
- 105 페이지 “시스템 전원 서비스 관리”

### 시스템 콘솔, 터미널 장치 및 전원 서비스를 관리하기 위한 새로운 기능

다음과 같이 새 기능 또는 변경된 기능이 Oracle Solaris 11에서 제공됩니다.

#### 시스템 전원 서비스 관리 방법의 변경 사항

전원 관리 구성이 SMF 구성 저장소로 이동했습니다. 전원 관련 명령, 데몬 및 구성 파일을 조합하여 사용하는 대신에 `poweradm` 명령을 사용하여 시스템 전원 관리 등록 정보를 직접 관리합니다. 이러한 변경 사항은 Oracle Solaris 11에서 현대적인 전원 관리 프레임워크를 구현하는 더 광범위한 변경 사항 세트의 일부입니다. 자세한 내용은 105 페이지 “시스템 전원 서비스 관리”를 참조하십시오.

## 시스템 콘솔 및 로컬로 연결된 터미널 장치 관리

시스템 콘솔은 특수한 속성을 갖고 있으며 특정 목적에 사용되는 터미널입니다. 예를 들면 관리자용의 커널 메시지는 콘솔에만 전송되고 다른 터미널에는 전송되지 않습니다.

터미널은 Oracle Solaris와 상호 작용하기 위한 수단입니다. 사용자 시스템의 비트맵 그래픽 디스플레이는 영숫자 터미널과 동일하지 않습니다. 영숫자 터미널은 직렬 포트에만 연결하며 텍스트만 표시합니다. 그래픽 디스플레이를 관리하기 위해 특별한 단계를 수행할 필요는 없습니다.

또한 터미널은 컴퓨터의 물리적 모니터 및 키보드 레이아웃과 연결될 수 있습니다. 그래픽 터미널의 경우는 컴퓨터의 그래픽 카드 및 모니터와 연결해야 한다는 차이가 있습니다. 따라서 문자가 직렬 포트에서 전송되는 대신에 컴퓨터에 있는 그래픽 카드의 메모리에 저장됩니다.

## 시스템 콘솔 및 로컬로 연결된 터미널 장치를 관리하는 SMF 서비스

시스템 콘솔 및 로컬로 연결된 터미널 장치는 SMF 서비스 `svc:/system/console`의 인스턴스로 표시됩니다. 이 서비스는 대부분의 동작을 정의하며 각 인스턴스 서비스에서 상속된 설정을 대체하는 특정 설정이 있습니다. `ttymon` 프로그램은 이러한 터미널에 로그인 서비스를 제공하는 데 사용됩니다. 각 터미널은 `ttymon` 프로그램의 개별 인스턴스를 사용합니다. 서비스가 `ttymon` 프로그램에 전달한 명령줄 인수가 해당 동작을 제어합니다.

시스템과 함께 제공되는 서비스 인스턴스는 다음과 같습니다.

- `svc:/system/console-login:default`  
기본 인스턴스는 항상 `ttymon` 프로그램이 시스템 하드웨어 콘솔에 로그인을 제공한다는 것을 나타냅니다. 예는 103 페이지 “시스템 콘솔의 설정을 수정하는 방법”을 참조하십시오.
- `svc:/system/console-login:{vt2, vt3, vt4, vt5, vt6}`  
시스템의 가상 콘솔을 위한 추가 서비스 인스턴스가 제공됩니다. 가상 콘솔을 사용할 수 없는 경우 이러한 서비스는 자동으로 사용 안함으로 설정됩니다. 자세한 내용은 `vtdaemon(1M)` 매뉴얼 페이지를 참조하십시오.
- `svc:/system/console-login:{terma, termb}`  
`svc:/system/console-login:terma` 및 `svc:/system/console-login:termb` 서비스는 편의상 제공됩니다. 이러한 서비스를 사용하면 추가 `/dev/term/a` 및 `/dev/term/b` 포트에 대한 로그인 서비스를 설정할 수 있습니다. 이러한 서비스는 기본적으로 **사용 안함**으로 설정됩니다.

svc:system/console-login 서비스의 일부로 추가 서비스 인스턴스를 정의할 수 있습니다. 예를 들어, 지원해야 하는 /dev/term/f 장치가 있는 경우 svc:/system/console-login:termf를 인스턴스화하고 적절하게 구성할 수 있습니다.

## ▼ 시스템 콘솔의 설정을 수정하는 방법

시스템 콘솔 관리는 SMF가 수행합니다. svccfg 명령을 사용하여 시스템 콘솔 등록 정보를 설정합니다.

다음 절차는 svccfg 명령을 사용하여 콘솔 터미널 유형을 변경하는 방법을 보여줍니다.

### 1 root 역할을 말합니다.

**Oracle Solaris 11.1 관리: 보안 서비스의 “지정된 관리 권한을 사용하는 방법”을 참조하십시오.**

```
$ su -
Password:
#
```

### 2 svccfg 명령을 사용하여 변경할 서비스 인스턴스의 등록 정보를 설정합니다.

예를 들어, :default 서비스가 나타내는 시스템 콘솔에 대한 터미널 유형을 변경하려면 다음 명령을 입력합니다.

```
# svccfg -s svc:/system/console-login:default "setprop ttymon/terminal_type = xterm"
```



주의 - svc:/system/console-login 서비스의 터미널 유형을 변경하면 모든 인스턴스에 영향을 주므로 변경하지 않는 것이 좋습니다.

## ▼ 보조 터미널에서 로그인 서비스를 설정하는 방법

시스템의 /dev/term/a 또는 /dev/term/b 직렬 포트에 연결된 터미널의 경우 미리 정의된 서비스가 제공됩니다.

/dev/term/a에 대한 로그인 서비스를 사용으로 설정하려면 다음 절차를 수행합니다.

### 1 root 역할을 말합니다.

**Oracle Solaris 11.1 관리: 보안 서비스의 “지정된 관리 권한을 사용하는 방법”을 참조하십시오.**

### 2 다음과 같이 서비스 인스턴스를 사용으로 설정합니다.

```
# svcadm enable svc:/system/console-login:terma
```

### 3 서비스가 온라인 상태인지 확인합니다.

```
# svcs svc:/system/console-login:terma
```

서비스가 온라인 상태라는 것이 출력에 표시되어야 합니다. 서비스가 유지 관리 모드인 경우 서비스의 로그 파일에서 자세한 내용을 참조하십시오.

## ▼ 시스템 터미널에 대한 변조 속도를 설정하는 방법

이 절차는 콘솔에 대한 변조 속도를 설정하는 방법을 보여줍니다. x86 기반 시스템에서 콘솔 속도에 대한 지원은 플랫폼에 따라 다릅니다.

SPARC 기반 시스템에 대해 지원되는 콘솔 속도는 다음과 같습니다.

- 9600 bps
- 19200 bps
- 38400 bps

### 1 관리자로 전환합니다.

**Oracle Solaris 11.1 관리: 보안 서비스의 “지정된 관리 권한을 사용하는 방법”**을 참조하십시오.

### 2 eeprom 명령을 사용하여 사용 중인 시스템 유형에 적합한 변조 속도를 설정합니다.

```
# eeprom ttya-mode=baud-rate,8,n,1,-
```

예를 들어 x86 기반 시스템의 콘솔에 대한 변조 속도를 38400으로 변경하려면 다음과 같이 입력합니다.

```
# eeprom ttya-mode=38400,8,n,1,-
```

### 3 /etc/ttydefs 파일에서 콘솔 라인을 다음과 같이 변경합니다.

```
console baud-rate hupcl opost onlcr:baud-rate::console
```

### 4 사용 중인 시스템 유형에 대해 다음과 같이 추가로 변경합니다.

이러한 변경 사항은 플랫폼에 따라 다릅니다.

- **SPARC 기반 시스템:** /etc/driver/drv 디렉토리에 있는 options.conf 파일 버전에서 변조 속도를 변경합니다.

변조 속도를 9600으로 변경하려면 다음 명령을 사용합니다.

```
# 9600 :bd:
ttymodes="2502:1805:bd:8a3b:3:1c:7f:15:4:0:0:0:11:13:1a:19:12:f:17:16";
```

변조 속도를 19200으로 변경하려면 다음 명령을 사용합니다.

```
# 19200 :be:
ttymodes="2502:1805:be:8a3b:3:1c:7f:15:4:0:0:0:11:13:1a:19:12:f:17:16";
```

변조 속도를 38400으로 변경하려면 다음 명령을 사용합니다.

```
# 38400 :bf:
ttymodes="2502:1805:bf:8a3b:3:1c:7f:15:4:0:0:0:11:13:1a:19:12:f:17:16";
```

- **x86 기반 시스템:** BIOS 직렬 리디렉션이 사용 가능한 경우 콘솔 속도를 변경합니다.

## 시스템 전원 서비스 관리

Oracle Solaris 11 릴리스에서 전원 관리 구성은 SMF 구성 저장소로 이동했습니다. 전원 관련 명령, 데몬 및 구성 파일을 조합하여 사용하는 대신에 새 `poweradm` 명령을 사용하여 시스템 전원 관리 등록 정보를 직접 관리합니다. 이러한 변경 사항은 Oracle Solaris 11에서 현대적인 전원 관리 프레임워크를 구현하는 더 광범위한 변경 사항 세트의 일부입니다.

다음 전원 관리 기능은 더 이상 사용할 수 없습니다.

- `/etc/power.conf`
- `pmconfig` 및 `powerd`
- 장치 전원 관리

다음 등록 정보는 전원 관리 구성 요소에 대해 설명합니다.

- `administrative-authority` - Oracle Solaris 전원 관리를 위한 관리 제어 소스를 정의합니다. 이 등록 정보는 `none`, `platform`(기본값) 또는 `smf`로 설정할 수 있습니다. `platform`으로 설정한 경우 플랫폼의 전원 관리 명령에서 `time-to-full-capacity` 및 `time-to-minimum-responsiveness`의 값이 사용됩니다. `smf`로 설정한 경우 SMF에서 `time-to-full-capacity` 및 `time-to-minimum-responsiveness`의 값이 사용됩니다. 반대 경로에 있을 때 플랫폼 명령 또는 SMF 서비스 등록 정보로부터 `time-to-full-capacity` 또는 `time-to-minimum-responsiveness`를 설정하려고 시도하면 값이 무시됩니다.

`administrative-authority`가 `none`으로 설정되어 있으면 Solaris 인스턴스 내의 전원 관리가 해제됩니다.

- `time-to-full-capacity` - 시스템이 활성 상태를 유지하면서 용량 또는 응답이 적은 상태에서 전체 용량에 도달할 때까지 허용되는 최대 시간(마이크로초)을 정의합니다. 최대 시간에는 용량 또는 응답이 적은 상태에서부터 이 경계 내에서 일부 또는 모든 PM 기능을 사용하는 동안 시스템이 전체 용량에 도달하도록 허용된 시간(재프로비전 및 제공)이 포함됩니다.

`administrative-authority`의 기본 설정이 플랫폼으로 설정되었기 때문에 기본적으로 이 값은 플랫폼(예: `i86pc`)에서 가져옵니다.

또는 `administrative-authority`가 `smf`로 설정된 경우 SMF 전원 서비스에서 제공된 정의로부터 이 값을 가져옵니다. 설치 시 이 값은 정의되어 있지 않습니다. 이 등록 정보를 수정하려면 시스템의 작업 로드 또는 응용 프로그램 요구에 적합한 값을 고려해야 합니다.

- `time-to-minimum-responsiveness` - 시스템이 활성 상태로 돌아갈 때까지 허용되는 시간(밀리초)을 정의합니다. 이 매개변수는 `time-to-full-capacity` 제약 조건을 충족하는 데 필요한 최소 용량을 제공합니다. `administrative-authority`의 기본 설정이 플랫폼으로 설정되었기 때문에 기본적으로 이 매개변수 값은 플랫폼(예: `i86pc`)에서 가져옵니다.

또는 `administrative-authority`가 `smf`로 설정된 경우 SMF 전원 서비스에서 제공된 정의로부터 이 값을 가져옵니다. 설치 시 이 값은 정의되어 있지 않습니다. 이 등록 정보를 수정하려면 시스템의 작업 로드 또는 응용 프로그램 요구에 적합한 값을 고려해야 합니다.

몇 초 정도의 중간 값으로 설정하면 플랫폼의 하드웨어 구성 요소 또는 부속 시스템을 응답 속도가 느린 비활성 상태로 배치할 수 있습니다. 30초에서 몇 분 정도의 큰 값으로 설정하면 `suspend-to-RAM`과 같은 기법을 사용하여 전체 시스템을 일시 중지할 수 있습니다.

- `suspend-enable` – 기본적으로 Solaris 실행 시스템은 일시 중지 작업을 시도하도록 허용되지 않습니다. 이 등록 정보를 `true`로 설정하면 일시 중지 작업을 시도하도록 허용할 수 있습니다. `administrative-authority` 값은 이 등록 정보에 영향을 주지 않습니다.
- `platform-disabled` – `platform-disabled`가 `true`로 설정되어 있으면 플랫폼의 전원 관리가 사용 안함으로 설정된 것입니다. `false`(기본값)로 설정하면 전원 관리가 위의 등록 정보 값으로 제어됩니다.

다음 명령을 사용하면 전원 관리 상태를 간단히 요약해서 표시할 수 있습니다.

```
$ /usr/sbin/poweradm show
Power management is enabled with the hardware platform as the authority:
time-to-full-capacity set to 250 microseconds
time-to-minimum-responsiveness set to 0 milliseconds
```

다음 명령을 사용하면 모든 전원 관리 등록 정보를 표시할 수 있습니다.

```
$ /usr/sbin/poweradm list
active_config/time-to-full-capacity           current=250, platform=250
active_config/time-to-minimum-responsiveness current=0, platform=0
active_control/administrative-authority      current=platform, smf=platform
suspend/suspend-enable                       current=false
platform-disabled                            current=false
```

위 출력에서 `active_control/administrative-authority`는 두 개의 설정이 있는 구성 소스를 나타냅니다.

- `platform` – 전원 관리를 위한 구성이 플랫폼에서 제공됩니다. 이것이 기본값입니다.
- `smf` – `poweradm` 명령을 사용하여 다른 전원 관리 등록 정보를 설정할 수 있습니다.

위의 출력에서 `platform-disabled` 등록 정보는 플랫폼 전원 관리가 사용으로 설정되었음을 나타냅니다.

```
platform-disabled           current=false
```

자세한 내용은 `poweradm(1M)`을 참조하십시오.

#### 예 5-1 전원 관리 사용 및 사용 안함

이전에 시스템을 일시 중단 및 다시 시작하기 위해 `/etc/power.conf` 파일에서 S3 지원을 사용으로 설정한 경우 다음과 비슷한 `poweradm` 구문을 사용합니다.

## 예 5-1 전원 관리 사용 및 사용 안함 (계속)

```
# poweradm set suspend-enable=true
```

기본적으로 suspend-enable 등록 정보는 false로 설정됩니다.

전원 관리를 사용 안함으로 설정하려면 다음 구문을 사용합니다.

```
# poweradm set administrative-authority=none
```

다음 SMF 전원 관리 서비스를 사용 안함으로 설정해도 전원 관리가 사용 안함으로 설정되지는 않습니다.

```
online          Sep_02   svc:/system/power:default
```

일시 중지 및 재개를 사용 안함으로 설정하려면 다음 구문을 사용합니다.

```
# poweradm set suspend-enable=false
```

## 예 5-2 전원 관리 매개변수 설정 및 표시

다음 예제에서는 time-to-full-capacity를 300마이크로초로 설정하고 time-to-minimum-responsiveness를 500밀리초로 설정하는 방법을 보여줍니다. 마지막으로 Oracle Solaris 인스턴스는 새로운 값에 대한 알림을 받습니다.

```
# poweradm set time-to-full-capacity=300
# poweradm set time-to-minimum-responsiveness=500
# poweradm set administrative-authority=smf
```

다음 명령은 현재 time-to-full-capacity 값을 보여줍니다.

```
# poweradm get time-to-full-capacity
300
```

다음 명령은 플랫폼별로 설정된 time-to-full-capacity 값을 검색합니다.

```
# poweradm get -a platform time-to-full-capacity
```

administrative-authority가 플랫폼으로 설정된 경우 이 값은 현재 값과 동일합니다. 자세한 내용은 위의 administrative-authority 등록 정보 설명을 참조하십시오.

## 시스템 전원 문제 해결

### ▼ 유지 관리 모드의 전원 관리 서비스에서 복구하는 방법

time-to-full-capacity 및 time-to-minimum-responsiveness를 설정하기 전에 administrative-authority가 smf로 설정된 경우 서비스가 유지 관리 모드가 됩니다. 이 시나리오에서 복구하려면 아래의 단계를 참조하십시오.

- 1 관리자로 전환합니다.

**Oracle Solaris 11.1 관리: 보안 서비스의 “지정된 관리 권한을 사용하는 방법”을 참조하십시오.**

- 2 **administrative-authority**를 **none**으로 설정합니다.

```
# poweradm set administrative-authority=none
```

- 3 **time-to-full-capacity** 및 **time-to-minimum-responsiveness**를 둘 다 원하는 값으로 설정합니다.

```
# poweradm set time-to-full-capacity=value  
# poweradm set time-to-minimum-responsiveness=value
```

- 4 서비스를 지웁니다.

```
# svcadm clear power
```

- 5 **administrative-authority**를 **smf**로 설정합니다.

```
# poweradm set administrative-authority=smf
```

# 색인

---

## A

at.deny 파일, 95, 98

설명, 81

at 명령, 94, 95, 98

-l 옵션(목록), 97

-m 옵션(메일), 95, 96

개요, 81, 82, 94

액세스 거부, 98

액세스 제어, 95, 98

개요, 81

오류 메시지, 98

자동 예약, 85

at 작업 파일, 94, 97

만들기, 95, 96

삭제, 97

설명, 83

위치, 83

제출, 94

표시, 97

atjobs 디렉토리, 85

설명, 81

## C

CPU(중앙 처리 장치)

고사용 프로세스, 45

정보 표시

시간 사용, 30, 45

cron.allow 파일, 90, 91, 92

cron.deny 파일, 90, 91

기본값, 90

cron 데몬, 83, 85

crontab 명령, 91

cron 데몬, 85

-e 옵션(편집), 86, 87

-l 옵션(목록), 88

-r 옵션(제거), 89, 90

개요, 81, 82

변경 사항 저장 없이 종료, 87

사용된 파일, 85

액세스 제어, 90, 91, 92

개요, 81, 90, 91

액세스 거부, 90, 91

지정된 사용자로 액세스 제한, 92

특정 사용자로 액세스 제한, 90, 91

예약, 85

오류 메시지, 93

일별 작업, 82

crontab 파일

구문, 85, 86

기본값, 84

만들기, 86, 87

만들기 및 편집, 83-84

삭제, 89, 90

설명, 85

액세스 거부, 91

위치, 84

제거, 89-90

편집, 86, 87

표시, 88

**D**

df 명령, 57  
 -k 옵션(킬로바이트), 57  
 개요, 57  
 예, 57  
 dispadmin 명령, 개요, 39

**E**

eeprom 명령, ttymon 터미널에 대한 변조 속도를  
 설정하기 위해 사용, 104  
 /etc/cron.d/at.deny 파일, 95, 98  
 /etc/cron.d/cron.allow 파일, 90, 91, 92  
 /etc/cron.d/cron.deny 파일, 90, 91

**F**

fcntl 정보, 31, 33  
 fsck 명령, 82  
 fstat 정보, 31, 33

**I**

iostat 명령  
 개요, 55  
 기본 정보 표시, 55

**K**

klwp 구조, 49  
 kthread 구조, 49

**L**

LWP(Lightweight Process)  
 구조, 49  
 정의, 48  
 프로세스 및, 48, 49  
 LWP(경량 프로세스), 정보 표시, 31

**M**

MOTD(오늘의 메시지) 기능, 25  
 motd 파일, 25

**N**

nice 명령, 43, 44, 45  
 nice 번호, 30, 44

**P**

perf 파일, 77  
 pfiles 명령, 31, 33  
 pflags 명령, 31  
 pkill 명령, 31, 35  
 pldd 명령, 31  
 pmap 명령, 31  
 priocntl 명령  
 -c 옵션(예약 클래스 지정), 42  
 -i 옵션(ID 유형), 42  
 -l 옵션(예약 클래스 표시), 39  
 -m 옵션(최대/최소 우선 순위), 42  
 -p 옵션(우선 순위 지정), 42  
 -s 옵션(우선 순위 상한/우선 순위 변경), 42  
 개요, 39  
 /proc 디렉토리, 30  
 proc 구조, 30, 49  
 PROCFS(프로세스 파일 시스템), 30  
 prtconf 명령, 16  
 시스템의 제품 이름 표시, 15-16  
 ps 명령, 29, 32  
 -c 옵션(예약 클래스), 30, 45  
 -ecl 옵션(전역 우선 순위), 39  
 -ef 옵션(전체 정보), 32  
 개요, 29  
 보고되는 필드, 30  
 psig 명령, 31  
 pstack 명령, 31  
 ptime 명령, 31  
 ptree 명령, 31, 33  
 pwait 명령, 31  
 pwdx 명령, 31, 33

**R**

runaway 프로세스, 45

**S**

sa1 명령, 76

sa2 명령, 76, 77

sadc 명령, 76, 77

sadd 파일, 77

sar 명령, 60, 77

-A 옵션(전체 성능), 76, 78

-a 옵션(파일 액세스), 60, 61

-b 옵션(버퍼), 61

-c 옵션(시스템 호출), 63

-e 옵션(종료 시간), 78

-f 옵션(데이터를 추출할 파일), 78

-i 옵션(간격), 78

-m 옵션(프로세스간 통신), 68

-p 옵션(페이지 인/페이지 폴트), 69

-q 옵션(대기열), 70, 71

-r 옵션(사용되지 않은 메모리), 71

-s 옵션(시작 시간), 78

-u 옵션(CPU 사용), 72

-v 옵션(시스템 테이블), 73

-y 옵션(터미널 장치), 75

개요, 60, 77

나열된 옵션, 78

모든 옵션에 대한 설명, 78

svcadm enable system/sar:default 명령, 77

sys crontab, 77

**T**

ttymon 콘솔 터미널에 대한 변조 속도 설정,  
방법, 104

**U**

user 구조, 49

/usr/proc/bin 디렉토리, 30, 31

**V**

/var/adm/sa/sadd 파일, 77

/var/spool/cron/atjobs 디렉토리, 81, 83, 85

/var/spool/cron/crontabs 디렉토리, 84, 85

/var/spool/cron/crontabs/root 파일, 84

/var/spool/cron/crontabs/sys crontab, 77

vmstat 명령, 개요, 52

**공**

공유 메모리, 프로세스 가상 메모리, 49

**기**

기본값, nice 번호, 44

**나**

나열

실행 중인 프로세스, 32

프로세스, 32

**다**

다시 시작, 프로세스, 31

**도**

도구, 시스템 성능 모니터링, 50

**디**

디렉토리

프로세스에 대한 현재 작업 디렉토리, 31

디스크 공간

정보 표시

df 명령, 57

마운트 지점, 58

디스크 드라이브

- 오래된/비활성 파일 찾기 및 삭제, 87
- 정보 표시
- 사용 가능한 디스크 공간, 57

로

- 로그 파일, 자동으로 삭제, 87

루

- 루틴 작업을 자동으로 실행(개요), 81

만

만들기

- at 작업, 95
- at 작업, 96
- crontab 파일, 86,87

메

메모리

- 가상
- 프로세스, 49
- 공유
- 프로세스 가상 메모리, 49
- 정보 표시 예, 16
- 프로세스 구조 및, 49

문

- 문제 해결
- 프로세스, 45

반

- 반복적 시스템 작업, 91

변

변경

- crontab 파일, 86
- 날짜, 25
- 예약 클래스, 42
- 우선 순위, 42,44
- 시간 공유 프로세스, 43,44

변조 속도

- eeprom 명령을 사용하여 설정하는 방법, 104
- ttymon 터미널에 대해 설정하는 방법, 104

보

보안

- at 명령, 95
- crontab 명령, 91

부

- 부트, 실행 sadc 명령, 77

사

- 사용자 모드 우선 순위, 38
- 사용자 프로세스
- 우선 순위, 38
- 우선 순위 변경, 43,44

삭

삭제

- at 작업, 97
- crontab 파일, 89,90
- 로그 파일, 87
- 오래된/비활성 파일, 82

새

- 새 기능, svcadm enable system/sar:default 명령, 77

**성**

## 성능

- 모니터링 도구, 50
- 보고, 60
- 시스템 작업 모니터링, 50, 60, 76
- 작업 데이터의 수동 수집, 60, 77
- 작업 데이터의 자동 수집, 76, 77
- 추적되는 작업, 50
- 파일 액세스, 60, 61
- 프로세스 관리, 31, 44, 48

**시**

## 시간

- CPU 사용, 30, 45
- 많은 양의 CPU 시간 누적 프로세스, 45
- 시간 공유 프로세스
  - 예약 매개 변수 변경, 42
  - 우선 순위
    - 개요, 38
    - 범위, 38
    - 변경, 42, 43, 44
- 시스템 리소스
  - 개요, 48
  - 모니터링, 95
  - 자동, 95
- 시스템 작업
  - “crontab 명령, at 명령”참조
  - 데이터의 수동 수집, 77
  - 데이터의 자동 수집, 76, 77
  - 예약
    - 반복적 작업, 82, 84
    - 일회성 작업, 82, 94
  - 자동으로 예약, 81
  - 추적된 작업 목록, 50
- 시스템 작업 실행 자동화, 81
- 시스템의 물리적 프로세서 유형 표시, psrinfo
  - p, 22
- 시스템의 제품 이름, prtconf 명령으로 표시, 15-16

**실**

- 실시간 프로세스, 클래스 변경, 42

**예**

## 예약

- “crontab 명령, at 명령”참조
- 반복적 시스템 작업, 82, 84
- 일회성 시스템 작업, 82, 94
- 예약 클래스, 37
  - 변경, 42
  - 우선 순위 레벨, 38
  - 우선 순위 레벨 및, 42
  - 우선 순위 변경, 42, 44
  - 정보 표시, 30, 39, 40
  - 지정, 41

**오**

## 오류 메시지

- at 명령, 98
- crontab 명령, 93

**우**

## 우선 순위(프로세스)

- 개요, 38, 44
- 변경, 42, 44
  - 시간 공유 프로세스, 42, 43, 44
- 사용자 모드 우선 순위, 38
- 예약 클래스 및, 42
- 전역
  - 정의, 38
  - 표시, 39
- 정보 표시, 30, 39
- 지정, 41, 42

**월**

- 월별 작업(crontab으로 예약), 82

**응**

- 응용 프로그램 스택, 48, 49

## 일

일별 작업(crontab으로 예약), 82

## 자

자동 시스템 작업 데이터 수집, 76,77

자동 시스템 작업 보고, 76,77

자동 시스템 작업 실행

    단일 작업, 94,95,98

    반복적 작업, 91,92

## 전

전역 우선 순위

    정의, 38

    표시, 39

## 제

제거,crontab 파일, 89-90

제어

    at 명령에 대한 액세스, 81

    at 명령에 액세스, 95,98

    crontab 명령에 대한 액세스

        개요, 81

    crontab 명령에 액세스, 91,92

    프로세스, 34

제품 이름 정보 표시,prtconf 명령, 15-16

## 주

주별 작업(crontab으로 예약), 82

주소 공간 맵, 31

## 중

중지,일시적으로 프로세스, 31

## 최

최대값,nice 번호, 44

최소값,nice 번호, 44

## 추

추적 플래그, 31

## 칩

칩 다중 스레드 기능을 확인하는 psrinfo 명령 옵션,  
psrinfo -p, 22

## 커

커널 스레드

    구조, 30,49

    예약, 30

## 코

코어 파일,자동으로 삭제, 94

## 콘

콘솔 터미널,변조 속도를 설정하는 방법, 104

콘솔 터미널 변조 속도,eeeprom 명령을 사용하여  
설정, 104

## 터

터미널,프로세스 제어, 30

## 틀

틀

    프로세스, 31

    프로세스 정보 표시, 30

**파**

## 파일

fstat 및 fcntl 정보 표시, 31,33  
 액세스 작업 확인, 60,61

## 파일 시스템

디스크 공간 사용, 57  
 마운트 지점, 58

**편**

## 편집

crontab 파일, 86,87

**표**

## 표시

at 작업, 97  
 crontab 파일, 88  
 LWP 정보, 31  
 날짜 및 시간, 13-14  
 시스템 작업 정보, 60,77  
 시스템 정보  
 명령, 14  
 시스템에 설치된 메모리, 16  
 연결된 라이브러리, 31  
 예약 클래스 정보, 30,39,40  
 우선 순위 정보, 30,39  
 프로세스 정보(방법), 33-34  
 호스트 ID, 14

**프**

프로그램, 디스크 종속성, 61

## 프로세스

nice 번호, 30,43,44,45  
 proc 툴 명령, 30  
 proc 툴 명령을 사용하여 정보 표시, 31  
 proc 툴을 사용하여 정보 표시, 30  
 runaway, 45  
 구조, 30,49  
 다시 시작, 31  
 문제 해결, 45

## 프로세스 (계속)

스택 추적, 31  
 신호 작업, 31  
 연결된 라이브러리, 31  
 열린 파일에 대한 fstat 및 fcntl 정보, 31,33  
 예약 클래스, 37  
 변경, 42  
 우선 순위 레벨, 38  
 우선 순위 레벨 및, 42  
 우선 순위 변경, 42,44  
 정보 표시, 30,39,40  
 지정, 41  
 용어, 48,49  
 우선 순위, 44  
 개요, 38,44  
 변경, 42,44  
 사용자 모드 우선 순위, 38  
 시간 공유 프로세스 우선 순위 변경, 42,43,44  
 예약 클래스, 38  
 예약 클래스 및, 42  
 전역 우선 순위, 38,39  
 정보 표시, 30,39  
 지정, 41,42  
 응용 프로그램 스레드 및, 48,49  
 일시적으로 중지, 31  
 정보 표시, 29  
 LWP, 31  
 priocntl 명령, 39  
 ps 명령, 29,32,39  
 실행 중인 프로세스 나열, 32  
 프로세스 나열, 32  
 정보 표시(방법), 33-34  
 정의, 48  
 제어, 34  
 종료, 31,35  
 주소 공간 맵, 31  
 추적 플래그, 31  
 툴 명령, 31  
 트리, 31,33  
 현재 작업 디렉토리, 31,33  
 프로세스 종료, 31,35  
 프로세스 파일 시스템(PROCFS), 30

