

강원혁신플랫폼

리눅스프로그래밍

프로세스





Linux에서 프로세스에 신호를 보내 프로세스의 동작을 종료하거나 조작할 수 있도록 하는 데 사용하는 명령어는 무엇인가요?

kill 명령어





학습 내용

- 1 /proc 디렉터리
- 2 ps 명령어
- 3 kill 명령어
- 4 nice 명령어

학습 목표

- 📄 /proc 디렉터리에 대해 설명할 수 있다.
- 📄 ps, kill, nice 명령어에 대해 설명할 수 있다.

강원혁신플랫폼

리눅스프로그래밍



/proc 디렉터리

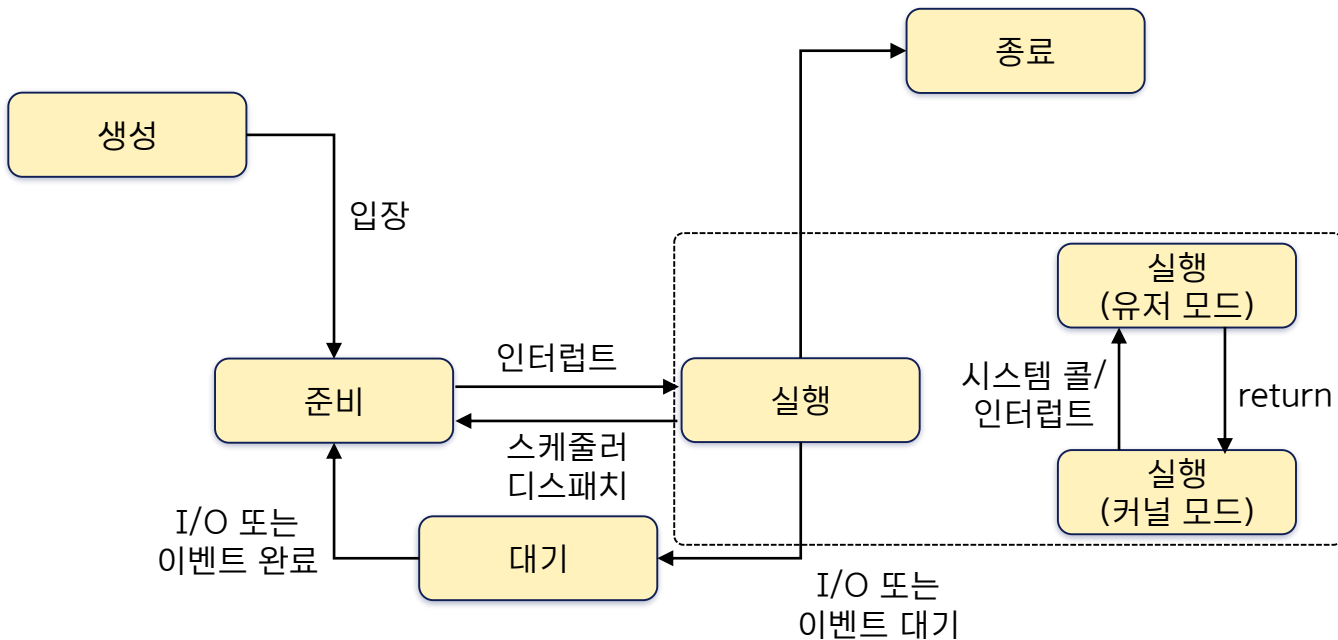




프로세스 컨트롤 블록

포인터	상태
프로세스 ID	
프로그램 카운터	
버퍼 포인터	
레지스터	
메모리 제한	
열린 파일 목록	
자식 링크	
....	

→ 생성, 준비, 대기, 실행, 종료





[

프로그램이 실행되어 프로세스가 생성되는 상태

]



프로세서에서 언제든지 **서비스의 실행을 기다리고 있는 상태로**
CPU의 할당을 기다리고 있음
실행 중에 시스템에서 특정 인터럽트가 발생하면 준비 상태로 전환됨



[코드가 현재 CPU에서 실행되고 있는 상태로 프로세서에 의해서
서비스를 받고 있는 상태임
커널에게 시스템 호출을 수행하게 할 수 있음]



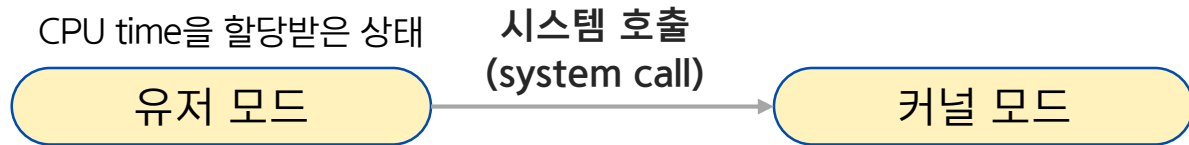
[커널에 시스템 호출을 처리하기 위해 기다리는 상태로
다른 프로세스가 서비스나 I/O 작업이 수행되고 있는 동안 대기함]



[

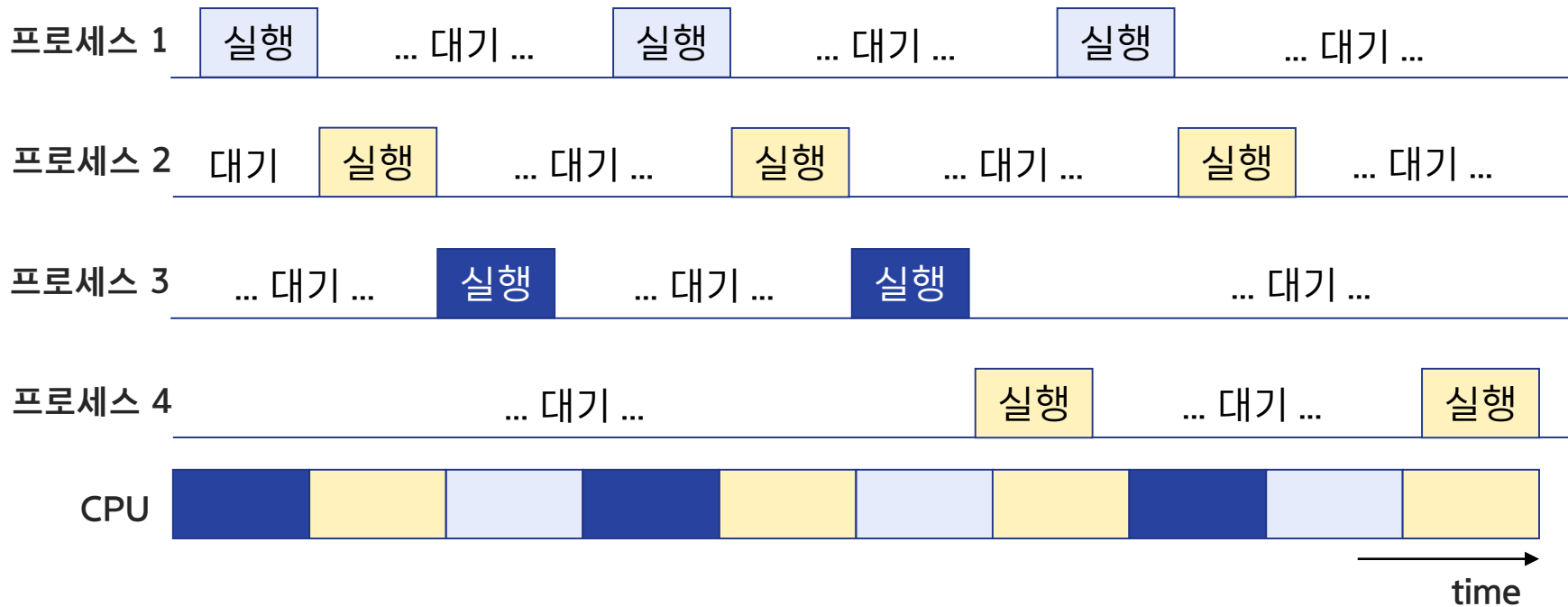
프로세스의 실행이 완료된 상태

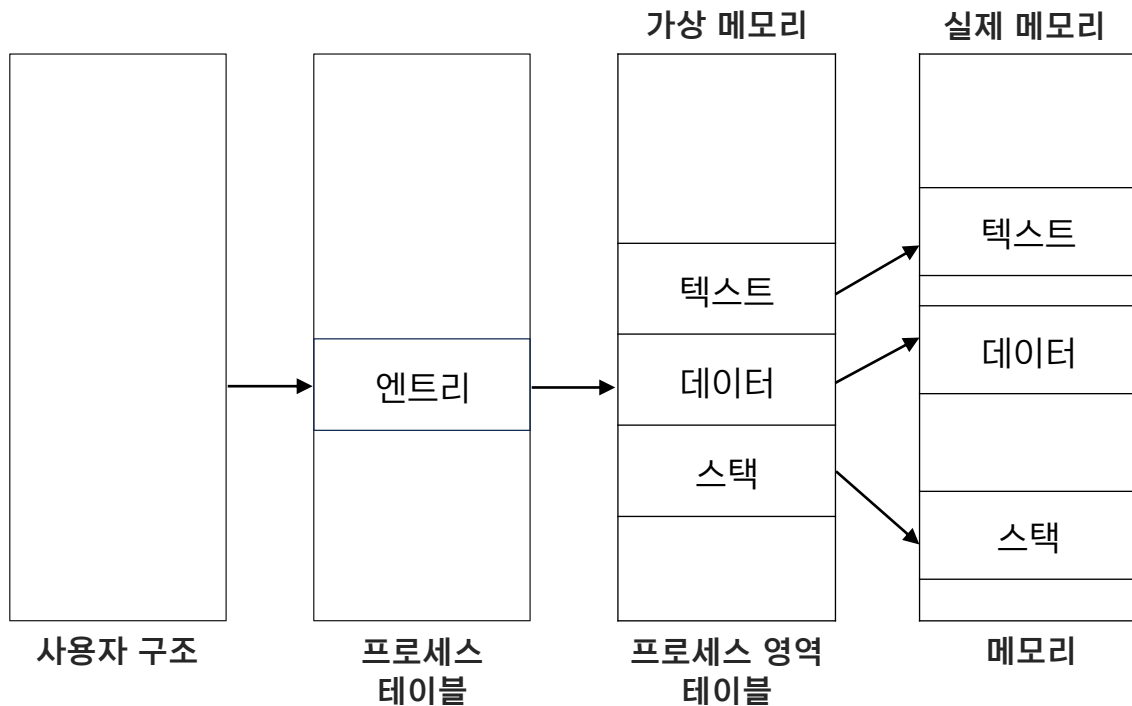
]



스와핑(swapping)

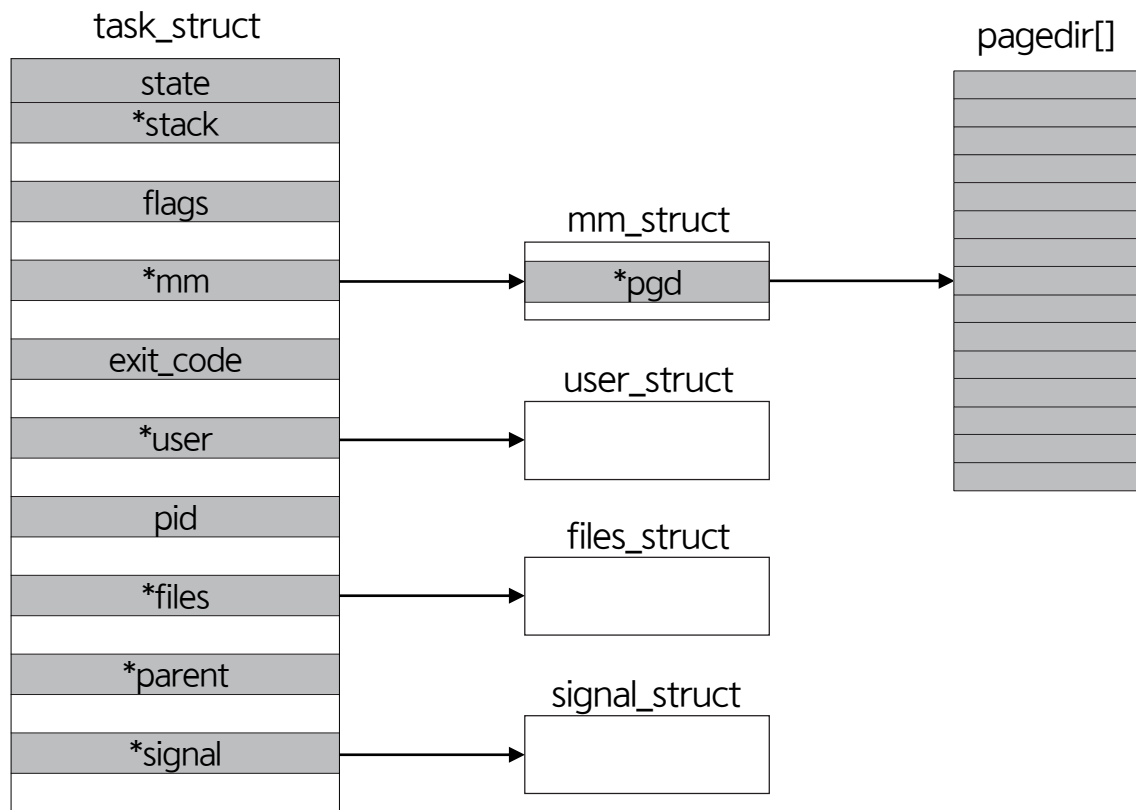
[대기/수면 상태가 지속될 때, 프로세스의 메모리 영역을
디스크의 가상 메모리공간으로 옮기는 동작 수행]







task_struct 자료형





특징

- ◆ 다양한 시스템 정보 및 프로세스 관련 데이터에 대한 인터페이스를 제공하는 특수 가상 파일 시스템
- ◆ /proc 디렉토리 내에서 시스템 및 실행 중인 프로세스의 다양한 측면을 나타내는 디렉토리 및 파일 확인



freetime@ATwin: ~/LinuxProg

```
$ ls /proc
```

1	197	388	64	92	cpuinfo	fs	keys	meminfo	schedstat	timer_list
111	2	399	72	acpi	crypto	interrupts	kmsg	misc	self	tty
113	200	40	74	buddyinfo	devices	iomem	kpagecgroup	modules	softirqs	uptime
125	201	413	79	bus	diskstats	ioports	kpagecount	mounts	stat	version
126	2495	452	835	cgroups	dma	irq	kpageflags	mtrr	swaps	vmallocinfo
127	324	454	84	cmdline	driver	kallsyms	loadavg	net	sys	vmstat
128	334	457	840	config.gz	execdomains	kcore	locks	pagetypeinfo	sysvipc	zoneinfo
188	383	5	85	consoles	filesystems	key-users	mdstat	partitions	thread-self	

```
$ |
```




\$ ps

PID	TTY	TIME	CMD
457	pts/2	00:00:00	bash
2496	pts/2	00:00:00	ps

\$ cat /proc/457/status

Name: bash

Umask: 0022

State: S (sleeping)

Tgid: 457

Ngid: 0

Pid: 457



PPid: 454

TracerPid: 0

Uid: 1000 1000 1000 1000

Gid: 1000 1000 1000 1000

FDSize: 256

Groups: 4 20 24 25 27 29 30 44 46 116 1000

NStgid: 457

NSpid: 457





VmPeak: 6352 kB
VmSize: 6320 kB
VmData: 1844 kB
VmStk: 132 kB
VmExe: 892 kB
VmLib: 1864 kB
VmPTE: 44 kB
VmSwap: 0 kB
HugetlbPages: 0 kB
CoreDumping: 0
THP_enabled: 1
Threads: 1





SigQ: 1/63223

NoNewPrivs: 0

Cpus_allowed: fffff

Cpus_allowed_list: 0-19

Mems_allowed: 1

Mems_allowed_list: 0

voluntary_ctxt_switches: 8816

nonvoluntary_ctxt_switches: 7

\$

강원혁신플랫폼

리눅스프로그래밍



ps 명령어





옵션	내용	전체 이름
u	프로세스의 사용자 이름과 시작 시간을 출력함	
f	프로세스의 정보를 한 줄로 자세히 출력함	(full)
l	프로세스의 정보를 한 줄로 보다 자세히 길게 출력함	(long)
e	환경에 대한 정보도 출력함	(environment)
a	다른 사용자들의 프로세스들을 모두 표시함	(all)



옵션	내용	전체 이름
x	로그인 상태에서 아직 완료되지 않은 프로세스들을 표시함 사용자가 종료한 후에도 일정 프로세스는 계속해서 실행되는데 터미널이 없는 상태에서 실행되는 프로세스들을 확인할 수 있음	
r	현재 실행 중인 프로세스들을 표시함	(running)
j	작업 중심의 형태로 출력함	(job)



구분	내용	비고
UID	사용자의 ID	(User ID)
USER	프로세스를 실행하고 있는 실제 사용자의 계정	
PID	프로세스의 ID	(Process ID)
PPID	부모 프로세스의 ID	(Parent Process ID)
STIME	프로세스 실행의 시작 시간	START
TTY	현재 프로세스의 표준 입출력을 담당하는 터미널	TT
TIME	프로세스에서 CPU를 점유하는 시간	
COMMAND	실행 명령어	CMD



구분	내용	비고
STAT	프로세스의 현재 상태	S
PRI	스케줄링에서 프로세스의 우선순위	(PRIority)
%CPU	최근 1분 동안 프로세서가 사용한 CPU 시간의 백분율	
RSS	프로세서에서 사용하는 실제 메모리의 용량(Kbyte)	(Resident Set Size)
ADDR	프로세스의 시작 메모리 주소	
SZ	프로세스가 사용하는 자료와 스택의 크기(Kbyte)	

강원혁신플랫폼

리눅스프로그래밍



kill 명령어





```
$ kill [-9][-15] {프로세스_ID}*
```

```
[-9] : SIGKILL
```

```
[-15] : SIGTERM
```

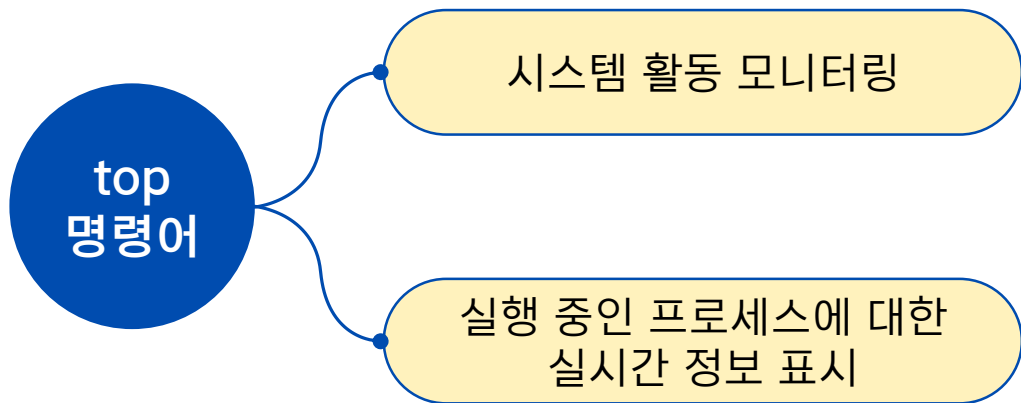


Kill 명령어

[

프로세스에 신호를 보내 프로세스의 동작을
종료하거나 조작할 수 있도록 하는 데 사용

]



```
freetime@ATwin: ~/LinuxProg x + v
top - 16:36:29 up 5:51, 1 user, load average: 0.05, 0.01, 0.00
Tasks: 33 total, 1 running, 32 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 15812.3 total, 14353.3 free, 567.8 used, 891.3 buff/cache
MiB Swap: 4096.0 total, 4096.0 free, 0.0 used. 14973.6 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
413	root	20	0	43764	38256	10140	S	0.3	0.2	0:21.97	python3
1	root	20	0	165732	11232	8348	S	0.0	0.1	0:00.33	systemd
2	root	20	0	2324	1256	1148	S	0.0	0.0	0:00.00	init-systemd(Ub
5	root	20	0	2352	80	68	S	0.0	0.0	0:00.00	init
40	root	19	-1	47804	15756	14720	S	0.0	0.1	0:00.11	systemd-journal
64	root	20	0	21828	5516	4212	S	0.0	0.0	0:00.11	systemd-udev
72	root	20	0	4784	1668	1228	S	0.0	0.0	0:00.60	snapfuse
74	root	20	0	4492	208	56	S	0.0	0.0	0:00.00	snapfuse
79	root	20	0	4624	172	16	S	0.0	0.0	0:00.00	snapfuse
84	root	20	0	4808	1812	1220	S	0.0	0.0	0:02.17	snapfuse
85	root	20	0	4768	1744	1276	S	0.0	0.0	0:00.69	snapfuse
92	systemd+	20	0	25260	12328	8388	S	0.0	0.1	0:00.08	systemd-resolve
111	root	20	0	4304	2624	2368	S	0.0	0.0	0:00.02	cron
113	message+	20	0	8592	4652	4072	S	0.0	0.0	0:00.02	dbus-daemon
125	root	20	0	30124	18480	9864	S	0.0	0.1	0:00.05	networkd-dispat
126	syslog	20	0	222400	7076	4524	S	0.0	0.0	0:00.01	rsyslogd
127	root	20	0	1466204	53068	19412	S	0.0	0.3	0:01.21	snappd
128	root	20	0	15324	7264	6324	S	0.0	0.0	0:00.07	systemd-logind
188	root	20	0	4780	3276	3036	S	0.0	0.0	0:00.03	subiquity-serve
197	root	20	0	3236	1116	1024	S	0.0	0.0	0:00.00	agetty
200	root	20	0	107212	21252	12984	S	0.0	0.1	0:00.03	unattended-upgr
201	root	20	0	3192	1076	988	S	0.0	0.0	0:00.00	agetty
324	root	20	0	548524	78816	22224	S	0.0	0.5	0:14.15	python3.10



단축키	내용	비고
M	메모리 사용량이 큰 순서로 정렬됨	shift + m
P	CPU 사용량이 큰 순서로 정렬됨	shift + p
T	실행 시간이 큰 순서로 정렬됨	shift + t
F	정렬할 내용을 선택할 수 있는 화면이 표시되는데 여기에서 표시될 내용을 선택할 수 있음	shift + f
R	정렬의 순서(올림차순/내림차순)를 바꿀 수 있음	shift + r
k	프로세스 kill - k 입력 후 종료할 PID 입력 signal을 입력하라고 하면 kill signal인 9를 입력함	



단축키	내용	비고
c	명령 인자 표시/비표시	
d	갱신 주기를 다시 설정함(기본 갱신 주기는 3초)	
스페이스바	다시 화면을 갱신함(refresh)	
u	설정한 사용자의 프로세스만 표시	which u

강원혁신플랫폼

리눅스프로그래밍



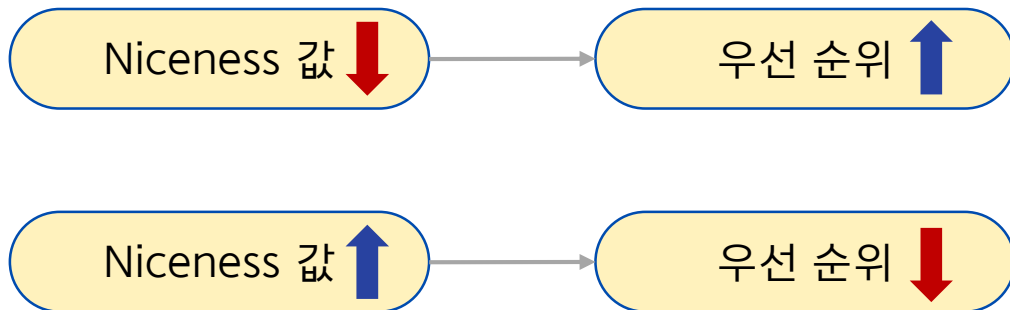
nice 명령어





특징

- ◆ 다른 프로세스와 관련된 프로세스의 일정에 영향을 주는 특정 우선 순위로 명령을 시작
- ◆ Niceness(-20~+19) 값을 조정하여 프로세스의 스케줄링 우선순위 조절





```
#include <stdio.h>
#include <unistd.h>

int main(int argc, char **argv)
{
    int i;

    /* 1초마다 0부터 숫자를 증가시키며 출력한다. */
    for(i = 0; ; i++) {
        printf("%10d\r", i);
        fflush(NULL);
        sleep(1);
    }
}
```



```
}
```

```
return 0;
```

```
}
```



```
$ gcc -o loop loop.c
```

무한 루프 실행, ^C로 종료

```
$ ./loop
```

```
^C 3
```

무한 루프 실행, 배경 ps로 설정

```
$ ./loop &
```

```
[1] 2528
```

```
12
```

직전의 배경 ps를 전경 ps로 가져옴

```
$ fg %1
```



```
./loop
```

```
14
```

```
^Z으로 프로세스 중단
```

```
^Z 23
```

```
[1]+ Stopped ./loop
```

```
bg로 배경 프로세스로 전환
```

```
$ bg
```

```
[1]+ ./loop &
```

```
26
```



단축키	내용	비고
Ctrl + d	현재 입력을 중지하거나 로그아웃을 수행함	
Ctrl + z	현재 실행 중인 프로세스를 일시 중단함	
Ctrl + c	현재 실행 중인 프로세스를 취소함	
Ctrl + l	화면을 지움	clear 명령어
Ctrl + s	키보드 잠금(lock)을 수행함	
Ctrl + q	키보드 잠금을 해제함	
Ctrl + p	이전 명령어를 표시함	ctrl + n
Shift + PageUp/PageDown	이미 지나간 앞/뒤의 내용을 표시함	
Alt + F1~F6	가상 터미널을 사용함	X 윈도우에서는 ctrl + Alt + F1~F6
Alt + F7~F12	가상 X 윈도우를 사용함	



01 • /proc 디렉터리

02 • ps 명령어

03 • kill 명령어

04 • nice 명령어