

## 응용과제 9

### <소스코드>

```
#define _GNU_SOURCE
#include<stdio.h>
#include<stdlib.h>
#include<sys/utsname.h>
#include<sys/sysinfo.h>
#include<sys/types.h>
#include<unistd.h>

void print_uts();
void print_sysinfo();
void print_sysconf();
void getdelim_file();

int main(int argc, char *argv[])
{
    int opt;
    struct utsname uts;
    if(argc<2)
    {
        fprintf(stderr, "이 프로그램은 파일명과 옵션을 사용해야합니다.\n"
            "사용법 : %s -(옵션)\n"
            "-a 모든정보 출력(cpu세부정보 제외) \n-m 머신타입 출력\n-n 노드이름 출력\n"
            "-o 운영체제 정보 출력 \n-r 운영체제 릴리즈 버전 출력 \n"
            "-s 운영체제 이름 출력 \n-v 운영체제 정보출력 \n-c cpu정보 출력\n",
            argv[0]);
        exit(1);
    }
    if(uname(&uts) == -1)
    {
        perror("uname");
        exit(2);
    }

    while((opt = getopt(argc, argv, "auifmnorsvc")) != -1)
    {
        switch(opt)
        {
            case 'a': //print all
                print_uts();
                print_sysinfo();
                print_sysconf();
                printf("\n===== 운영체제 정보 =====\n");
                getdelim_file("/etc/os-release", "0");
                printf("===== 운영체제 릴리즈 버전 =====\n");
                printf("Release : %s\n", uts.release);
```

```

        getdelim_file("/etc/redhat-release", "0");
        printf("==== 운영체제 정보 =====\n");
        getdelim_file("/proc/version", "1");
        break;
case 'u': //uname구조체
        print_uts();
        break;
case 'i': //시스템 통계정보
        print_sysinfo();
        break;
case 'f': //시스템자원정보
        print_sysconf();
        break;
case 'm': //시스템 하드웨어타입 (machine)
        printf("==== 머신(하드웨어)타입 =====\n");
        printf("Machine : %s\n", uts.machine);
        break;
case 'n': //노드이름 (nodename)
        printf("==== 노드이름 =====\n");
        printf("Node name : %s\n", uts.nodename);
        break;
case 'o': //운영체제정보 (OS)
        printf("==== 운영체제 정보 =====\n");
        getdelim_file("/etc/os-release", "0");
        break;
case 'r': //커널배포버전(kernel release)
        printf("==== 운영체제 릴리즈 버전 =====\n");
        printf("Release : %s\n", uts.release);
        getdelim_file("/etc/redhat-release", "1");
        break;
case 's': //운영체제 이름 (sys name)
        printf("==== 운영체제 이름 =====\n");
        printf("OS name : %s\n", uts.sysname);
        break;
case 'v': //운영체제 버전정보 (kernel version)
        printf("==== 운영체제 정보 =====\n");
        getdelim_file("/proc/version", "1");
        break;
case 'c': //cpu 정보 (cpu)
        printf("==== cpu 정보 =====\n");
        getdelim_file("/proc/cpuinfo", "1");
        break;

```

```

    }

```

```

}

```

```

return 0;

```

```

}

```

```

void print_uts(void)
{
    struct utsname uts;

    if(uname(&uts) == -1)
    {
        perror("uname");
        exit(2);
    }

    printf("OSname : %s\n", uts.sysname);
    printf("Nodename : %s\n", uts.nodename);
    printf("Release : %s\n", uts.release);
    printf("Version : %s\n", uts.version);
    printf("Machine : %s\n", uts.machine);
}

```

```

void print_sysinfo(void)
{
    struct sysinfo sys;

    sysinfo(&sys);
    printf("부팅 이후 지나간 초단위 시간 : %ld\n", sys.uptime);
    printf("1, 5, 15 분 부하 평균 : %lu\n", sys.loads);
    printf("사용할 수 있는 총 메모리 크기 : %lu\n", sys.totalram);
    printf("이용할 수 있는 메모리 크기 : %lu\n", sys.freeram);
    printf("공유 메모리의 양 : %lu\n", sys.sharedram);
    printf("버퍼에 의해 사용된 메모리 : %lu\n", sys.bufferram);
    printf("총 스왑 공간의 크기 : %lu\n", sys.totalswap);
    printf("이용할 수 있는 스왑 공간 : %lu\n", sys.freeswap);
    printf("현재 프로세스의 수 : %hd\n", sys.procs);
}

```

```

void print_sysconf(void)
{
    printf("argv[]와 envp[]를 합한 최대길이 : %ld \n", sysconf(_SC_ARG_MAX));
    printf("한 UID에 허용되는 최대 프로세스 개수 : %ld \n", sysconf(_SC_CHILD_MAX));
    printf("초당 클럭 틱 수 : %ld \n", sysconf(_SC_CLK_TCK));
    printf("프로세스당 열 수 있는 최대파일 개수 : %ld \n", sysconf(_SC_OPEN_MAX));
    printf("패스워드의 최대길이 : %ld \n", sysconf(_SC_PASS_MAX));
    printf("로그인명의 최대길이 : %ld \n", sysconf(_SC_LOGIN_NAME_MAX));
    printf("시스템 메모리의 페이지 크기 : %ld \n", sysconf(_SC_PAGESIZE));
}

```

```

void getdelim_file(char *path, int opt)
{
    FILE *fp;

```

```
while((fp=fopen(path,"r")) == NULL)
{
    perror("fopen");
    exit(3);
}

int delim;
switch(opt)
{
    case '0' : //getline처럼 동작
        delim = '\n';
        break;
    case '1' :
        delim = 0;
        break;
}

size_t size = 0;
char *lineptr = NULL;
while(getdelim(&lineptr, &size, delim, fp) != -1)
{ //lineptr가 널, size가 0으로 설정되어있으면 getdelim에서 버퍼를 할당
    puts(lineptr);
}
free(lineptr); //할당된 메모리 해제
fclose(fp);
}
```

include, 함수선언, main

```
#define _GNU_SOURCE
#include<stdio.h>
#include<stdlib.h>
#include<sys/utsname.h>
#include<sys/sysinfo.h>
#include<sys/types.h>
#include<unistd.h>

void print_uts();
void print_sysinfo();
void print_sysconf();
void getdelim_file();

int main(int argc, char *argv[])
{
    int opt;
    struct utsname uts;
    if(argc<2)
    {
        fprintf(stderr, "이 프로그램은 파일명과 옵션을 사용해야 합니다.\n"
            "사용법 : %s -(옵션)\n"
            "-a 모든 정보 출력 (cpu 세부 정보 제외) \n-m 머신 타입 출력 \n-n 노드 이름 출력 \n"
            "-o 운영체제 정보 출력 \n-r 운영체제 릴리즈 버전 출력 \n"
            "-s 운영체제 이름 출력 \n-v 운영체제 정보 출력 \n-c cpu 정보 출력 \n",
            argv[0]);

        exit(1);
    }
    if(uname(&uts) == -1)
    {
        perror("uname");
        exit(2);
    }

    while((opt = getopt(argc, argv, "auicmnorsvc")) != -1)
    {
        switch(opt)
        {
            case 'a': //print all
                print_uts();
                print_sysinfo();
                print_sysconf();
                printf("\n===== 운영체제 정보 =====\n");
                getdelim_file("/etc/os-release", "0");
                printf("===== 운영체제 릴리즈 버전 =====\n");
                printf("Release : %s\n", uts.release);
                getdelim_file("/etc/redhat-release", "0");
                printf("===== 운영체제 정보 =====\n");
                getdelim_file("/proc/version", "1");
                break;
```

□

```

case 'u': //uname구조체
    print_uts();
    break;
case 'i': //시스템 통계 정보
    print_sysinfo();
    break;
case 'f': //시스템 자원 정보
    print_sysconf();
    break;
case 'm': //시스템 하드웨어 타입 (machine)
    printf("===== 머신 (하드웨어) 타입 =====\n");
    printf("Machine : %s\n", uts.machine);
    break;
case 'n': //노드 이름 (nodename)
    printf("===== 노드 이름 =====\n");
    printf("Node name : %s\n", uts.nodename);
    break;
case 'o': //운영체제 정보 (OS)
    printf("===== 운영체제 정보 =====\n");
    getdelim_file("/etc/os-release", "0");
    break;
case 'r': //커널 배포 버전 (kernel release)
    printf("===== 운영체제 릴리즈 버전 =====\n");
    printf("Release : %s\n", uts.release);
    getdelim_file("/etc/redhat-release", "1");
    break;
case 's': //운영체제 이름 (sys name)
    printf("===== 운영체제 이름 =====\n");
    printf("OS name : %s\n", uts.sysname);
    break;
case 'v': //운영체제 버전 정보 (kernel version)
    printf("===== 운영체제 정보 =====\n");
    getdelim_file("/proc/version", "1");
    break;
case 'c': //cpu 정보 (cpu)
    printf("===== cpu 정보 =====\n");
    getdelim_file("/proc/cpuinfo", "1");
    break;
    }
}
return 0;
}

```

getdelim을 사용하기위해 \_GNU\_SOURCE를 정의해주었습니다. utsname구조체와 sysinfo구조체를 사용하여 정보를 출력하기위해 헤더를 인클루드 해주었습니다.

그 후 함수를 선언해주었고 utsname구조체 uts를 선언하였습니다.

argc가 2 미만인경우를 조건으로 사용하여 파일명과 옵션을 받도록 하였고, 조건을 만족하지 못할 경우 에러메세지와 사용법을 출력해주었습니다.

uname함수를 사용하여 uts구조체에 운영체제 정보를 저장하였습니다.그후 getopt루프문을 통하여 각 옵션 입력시에 해당 옵션으로 동작하도록 하였습니다.

uts,sysinfo,sysconf함수 출력부분을 따로 void형 함수로 만들어서 출력해주었습니다.

m,n,r,s 는 uts구조체의 각 멤버에 접근하여 정보를 출력해주었습니다.

getdelim\_file함수를 이용한 출력은 후에 함수설명쪽에서 설명하겠습니다.

uts, sysinfo, sysconf 출력함수

```
void print_uts(void)
{
    struct utsname uts;

    if(uname(&uts) == -1)
    {
        perror("uname");
        exit(2);
    }

    printf("OSname : %s\n", uts.sysname);
    printf("Nodename : %s\n", uts.nodename);
    printf("Release : %s\n", uts.release);
    printf("Version : %s\n", uts.version);
    printf("Machine : %s\n", uts.machine);
}

void print_sysinfo(void)
{
    struct sysinfo sys;

    sysinfo(&sys);
    printf("부팅 이후 지나간 초단위 시간 : %ld\n", sys.uptime);
    printf("1, 5, 15 분 부하 평균 : %lu\n", sys.loads);
    printf("사용할 수 있는 총 메모리 크기 : %lu\n", sys.totalram);
    printf("이용할 수 있는 메모리 크기 : %lu\n", sys.freeram);
    printf("공유 메모리의 양 : %lu\n", sys.sharedram);
    printf("버퍼에 의해 사용된 메모리 : %lu\n", sys.bufferram);
    printf("총 스왑 공간의 크기 : %lu\n", sys.totalswap);
    printf("이용할 수 있는 스왑 공간 : %lu\n", sys.freeswap);
    printf("현재 프로세스의 수 : %hd\n", sys.procs);
}

void print_sysconf(void)
{
    printf("argv[]와 envp[]를 합한 최대 길이 : %ld\n", sysconf(_SC_ARG_MAX));
    printf("한 UID에 허용되는 최대 프로세스 개수 : %ld\n", sysconf(_SC_CHILD_MAX));
    printf("초당 클럭 틱 수 : %ld\n", sysconf(_SC_CLK_TCK));
    printf("프로세스당 열 수 있는 최대 파일 개수 : %ld\n", sysconf(_SC_OPEN_MAX));
    printf("패스워드의 최대 길이 : %ld\n", sysconf(_SC_PASS_MAX));
    printf("로그인명의 최대 길이 : %ld\n", sysconf(_SC_LOGIN_NAME_MAX));
    printf("시스템 메모리의 페이지 크기 : %ld\n", sysconf(_SC_PAGESIZE));
}
```

출력함수부분입니다. sys/utsname.h 헤더에 정의되어있는 utsname구조체 uts 를 선언하여 uname함수를 호출하고 uts구조체에 저장하여 출력하는식으로 os이름, 노드이름, 릴리즈번호, 버전, 머신 멤버를 출력하였습니다.

sys/sysinfo.h 헤더에 정의되어있는 sysinfo구조체 sys를 선언하여 sysinfo함수를 호출하고 정보를 sys구조체에 저장하여 출력해주었습니다. 이부분에서 lily는 linux기반의 cent os이기에 유닉스의 sysinfo함수와는 조금 다르기에 man sysinfo를 통해 사용 매뉴얼을 확인해주었습니다. 각 구조체가 long, unsigned long, unsigned short로 정의되어있어서 %ld, %lu, %hd로 출력해주었습니다.

sysconf함수를 사용하여 시스템의 자원정보를 출력해주었습니다.



## 파일 열기 및 출력함수 getdelim\_file

```
void getdelim_file(char *path, int opt)
{
    FILE *fp;
    while((fp=fopen(path,"r")) == NULL)
    {
        perror("fopen");
        exit(3);
    }

    int delim;
    switch(opt)
    {
        case '0' : //getline처럼 동작
            delim = '\n';
            break;
        case '1' :
            delim = 0;
            break;
    }

    size_t size = 0;
    char *lineptr = NULL;
    while(getdelim(&lineptr, &size, delim, fp) != -1)
    { //lineptr가 널, size가 0으로 설정되어 있으면 getdelim에서 버퍼를 할당
        puts(lineptr);
    }
    free(lineptr); //할당된 메모리 해제
    fclose(fp);
}
```

이함수는 파일경로를 받아서 해당 파일을 출력해주는 함수입니다. getdelim함수를 사용하여 구분자단위로 잘라내어 출력하였습니다. getdelim함수를 사용하면 버퍼메모리가 부족할 경우 알아서 realloc을 호출하여 버퍼크기를 늘려주기 때문에 메모리할당, 재할당, 오버플로우 체크등의 작업을 하지 않아도 된다는점이 장점입니다. /proc디렉토리의 출력에 사용하려는 파일들은 0을 구분자로 사용하기 때문에 getline으로 출력하게되면 개행문자가 두 번 출력됩니다. 따라서 해당 디렉토리의 파일들은 delim에 0을 할당하여 사용해주었고 나머지는 getline처럼 사용되면 되기 때문에 구분자로 Wn을 할당하여 사용해주었습니다. 후에 free함수로 메모리를 해제해주었습니다.

따로 옵션을 선택하여 정보를 출력할 때 os의 정보등을 출력하기위해서 함수와 시스템콜등을 찾아보았는데 uts구조체를 사용하여 출력하는 간단한 정보정도밖에 출력되지않아 어떻게하면 좀더 세부적인 내용을 출력할 수 있을까 생각해보았습니다. 그러던중 유닉스 디렉토리의 구성을 보면 /etc는 시스템 환경설정 디렉토리, /proc은 커널과 프로세스를 위한 파일시스템 디렉토리임을 확인하게되었고 이 디렉토리를 찾아보니 해당 내용들을 확인 할 수 있었습니다. 커널버전 proc/version, 배포판 버전확인 /etc/redhat-release, 메모리정보확인 /proc/meminfo, 운영체제 정보 /etc/os-release 등등...을 cat명령어로 찾아보니 조금더 디테일한 내용을 확인 할 수 있었습니다.

확인한 파일들 메인에서 경로를 받아 a,o,r,v,c옵션을 사용할 때 출력해주었습니다.

## 옵션 없이 진행

```
[khm970514@lily ch4]$ gcc -o my_uname my_uname.c
```

```
[khm970514@lily ch4]$ ./my_uname
```

이 프로그램은 파일명과 옵션을 사용해야 합니다.

사용법 : ./my\_uname -(옵션)

-a 모든 정보 출력 (cpu세부 정보 제외)

-m 머신 타입 출력

-n 노드 이름 출력

-o 운영체제 정보 출력

-r 운영체제 릴리즈 버전 출력

-s 운영체제 이름 출력

-v 운영체제 정보 출력

-c cpu 정보 출력

## 옵션 -a

```
[khm970514@lily ch4]$ gcc -o my_uname my_uname.c
[khm970514@lily ch4]$ ./my_uname -a
OSname : Linux
Nodename : lily.mmu.ac.kr
Release : 3.10.0-957.12.1.el7.x86_64
Version : #1 SMP Mon Apr 29 14:59:59 UTC 2019
Machine : x86_64
부팅 이후 지나간 초 단위 시간 : 14620336
1, 5, 15 분 부하 평균 : 140736987371768
사용할 수 있는 총 메모리 크기 : 16612028416
이용할 수 있는 메모리 크기 : 6240628736
공유 메모리의 양 : 0
버퍼에 의해 사용된 메모리 : 5361664
총 스왑 공간의 크기 : 8464101376
이용할 수 있는 스왑 공간 : 7912427520
현재 프로세스의 수 : 495
argv[]와 envp[]를 합한 최대 길이 : 2097152
한 UID에 허용되는 최대 프로세스 개수 : 4096
초당 클럭 틱 수 : 100
프로세스당 열 수 있는 최대 파일 개수 : 1024
패스워드의 최대 길이 : 8192
로그인명의 최대 길이 : 256
시스템 메모리의 페이지 크기 : 4096
```

===== 운영 체제 정보 =====

```
NAME="CentOS Linux"
VERSION="7 (Core)"
ID="centos"
ID_LIKE="rhel fedora"
VERSION_ID="7"
PRETTY_NAME="CentOS Linux 7 (Core)"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:centos:centos:7"
HOME_URL="https://www.centos.org/"
BUG_REPORT_URL="https://bugs.centos.org/"

CENTOS_MANTISBT_PROJECT="CentOS-7"
CENTOS_MANTISBT_PROJECT_VERSION="7"
REDHAT_SUPPORT_PRODUCT="centos"
REDHAT_SUPPORT_PRODUCT_VERSION="7"
```

===== 운영 체제 릴리즈 버전 =====

```
Release : 3.10.0-957.12.1.el7.x86_64
CentOS Linux release 7.7.1908 (Core)
```

===== 운영 체제 정보 =====

```
Linux version 3.10.0-957.12.1.el7.x86_64 (mockbui
ld@kbuilder.bsys.centos.org) (gcc version 4.8.5 2
0150623 (Red Hat 4.8.5-36) (GCC) ) #1 SMP Mon Apr
29 14:59:59 UTC 2019
```

```
[khm970514@lily ch4]$ █
```

## 옵션 -u -i -f -m -n -o -r -s -v

```
[khm970514@lily ch4]$ ./my_uname -u
OSname : Linux
Nodename : lily.mmu.ac.kr
Release : 3.10.0-957.12.1.el7.x86_64
Version : #1 SMP Mon Apr 29 14:59:59 UTC 2019
Machine : x86_64
[khm970514@lily ch4]$ ./my_uname -i
부팅 이후 지나간 초단위 시간 : 14620410
1, 5, 15 분 부하 평균 : 140721595070840
사용할 수 있는 총 메모리 크기 : 16612028416
이용할 수 있는 메모리 크기 : 6241574912
공유 메모리의 양 : 0
버퍼에 의해 사용된 메모리 : 5361664
출스왑 공간의 크기 : 8464101376
이용할 수 있는 스왑 공간 : 7912427520
현재 프로세스의 수 : 494
[khm970514@lily ch4]$ ./my_uname -f
argv[]와 envp[]를 합한 최대 길이 : 2097152
한 UID에 허용되는 최대 프로세스 개수 : 4096
초당 클럭 틱 수 : 100
프로세스당 열 수 있는 최대 파일 개수 : 1024
패스워드의 최대 길이 : 8192
로그인명의 최대 길이 : 256
시스템 메모리의 페이지 크기 : 4096
[khm970514@lily ch4]$ ./my_uname -m
===== 머신 (하드웨어) 타입 =====
Machine : x86_64
[khm970514@lily ch4]$ ./my_uname -n
===== 노드 이름 =====
Node name : lily.mmu.ac.kr
[khm970514@lily ch4]$ ./my_uname -o
===== 운영체제 정보 =====
NAME="CentOS Linux"
VERSION="7 (Core)"
ID="centos"
ID_LIKE="rhel fedora"
VERSION_ID="7"
PRETTY_NAME="CentOS Linux 7 (Core)"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:centos:centos:7"
HOME_URL="https://www.centos.org/"
BUG_REPORT_URL="https://bugs.centos.org/"

CENTOS_MANTISBT_PROJECT="CentOS-7"
CENTOS_MANTISBT_PROJECT_VERSION="7"
REDHAT_SUPPORT_PRODUCT="centos"
REDHAT_SUPPORT_PRODUCT_VERSION="7"

[khm970514@lily ch4]$ ./my_uname -r
===== 운영체제 릴리즈 버전 =====
Release : 3.10.0-957.12.1.el7.x86_64
CentOS Linux release 7.7.1908 (Core)

[khm970514@lily ch4]$ ./my_uname -s
===== 운영체제 이름 =====
OS name : Linux
[khm970514@lily ch4]$ ./my_uname -v
===== 운영체제 정보 =====
Linux version 3.10.0-957.12.1.el7.x86_64 (mockbui
ld@kbuilder.bsys.centos.org) (gcc version 4.8.5 2
0150623 (Red Hat 4.8.5-36) (GCC) ) #1 SMP Mon Apr
29 14:59:59 UTC 2019

[khm970514@lily ch4]$
```

## 옵션 -c

```
[khm970514@lily ch4]$ ./my_uname -c
===== cpu 정보 =====
processor      : 0
vendor_id     : GenuineIntel
cpu family    : 6
model         : 45
model name    : Intel(R) Xeon(R) CPU E5-2640 0
@ 2.50GHz
stepping      : 7
microcode     : 0x718
cpu MHz       : 1200.000
cache size    : 15360 KB
physical id    : 0
siblings      : 12
core id       : 0
cpu cores     : 6
apicid        : 0
initial apicid : 0
fpu           : yes
fpu_exception : yes
cpuid level   : 13
wp            : yes
flags         : fpu vme de pse tsc msr pae mce
cx8 apic sep mtrr pge mca cmov pat pse36 clflush
dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall n
x pdpe1gb rdtscp lm constant_tsc arch_perfmon peb
s bts rep_good nopl xtopology nonstop_tsc aperfmp
erf eagerfpu pni pclmulqdq dtes64 monitor ds_cpl
vmx smx est tm2 ssse3 cx16 xtpr pdcm pcid dca sse
4_1 sse4_2 x2apic popcnt tsc_deadline_timer aes x
save avx lahf_lm epb ssbd ibrs ibpb stibp tpr_sha
dow vnmi flexpriority ept vpid xsaveopt dtherm id
a arat pln pts spec_ctrl intel_stibp flush_l1d
bogomips      : 4987.73
clflush size   : 64
cache_alignment : 64
address sizes  : 46 bits physical, 48 bits virtu
al
power management:

processor      : 1
vendor_id     : GenuineIntel
cpu family    : 6
model         : 45
model name    : Intel(R) Xeon(R) CPU E5-2640 0
@ 2.50GHz
stepping      : 7
microcode     : 0x718
cpu MHz       : 1200.000
cache size    : 15360 KB
physical id    : 0
siblings      : 12
core id       : 1
cpu cores     : 6
apicid        : 2
initial apicid : 2
fpu           : yes
fpu_exception : yes
cpuid level   : 13
wp            : yes
flags         : fpu vme de pse tsc msr pae mce
cx8 apic sep mtrr pge mca cmov pat pse36 clflush
dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall n
x pdpe1gb rdtscp lm constant_tsc arch_perfmon peb
s bts rep_good nopl xtopology nonstop_tsc aperfmp
erf eagerfpu pni pclmulqdq dtes64 monitor ds_cpl
vmx smx est tm2 ssse3 cx16 xtpr pdcm pcid dca sse
4_1 sse4_2 x2apic popcnt tsc_deadline_timer aes x
save avx lahf_lm epb ssbd ibrs ibpb stibp tpr_sha
dow vnmi flexpriority ept vpid xsaveopt dtherm id
a arat pln pts spec_ctrl intel_stibp flush_l1d
bogomips      : 4987.73
```



## 옵션 -uifmnorsv

```
[khm970514@lily ch4]$ ./my_uname -uifmnorsv
OSname : Linux
Nodename : lily.mmu.ac.kr
Release : 3.10.0-957.12.1.el7.x86_64
Version : #1 SMP Mon Apr 29 14:59:59 UTC 2019
Machine : x86_64
부팅 이후 지나간 초 단위 시간 : 14620748
1, 5, 15 분 부하 평균 : 140721982069432
사용할 수 있는 총 메모리 크기 : 16612028416
이용할 수 있는 메모리 크기 : 6241267712
공유 메모리의 양 : 0
버퍼에 의해 사용된 메모리 : 5361664
총 스왑 공간의 크기 : 8464101376
이용할 수 있는 스왑 공간 : 7912427520
현재 프로세스의 수 : 495
argv[]와 envp[]를 할한 최대 길이 : 2097152
한 UID에 허용되는 최대 프로세스 개수 : 4096
초당 클럭 틱 수 : 100
프로세스당 열 수 있는 최대 파일 개수 : 1024
패스워드의 최대 길이 : 8192
로그인명의 최대 길이 : 256
시스템 메모리의 페이지 크기 : 4096
===== 머신 (하드웨어)타입 =====
Machine : x86_64
===== 노드 이름 =====
Node name : lily.mmu.ac.kr
===== 운영 체제 정보 =====
NAME="CentOS Linux"
VERSION="7 (Core)"
ID="centos"
ID_LIKE="rhel fedora"
VERSION_ID="7"
PRETTY_NAME="CentOS Linux 7 (Core)"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:centos:centos:7"
HOME_URL="https://www.centos.org/"
BUG_REPORT_URL="https://bugs.centos.org/"

CENTOS_MANTISBT_PROJECT="CentOS-7"
CENTOS_MANTISBT_PROJECT_VERSION="7"
REDHAT_SUPPORT_PRODUCT="centos"
REDHAT_SUPPORT_PRODUCT_VERSION="7"

===== 운영 체제 릴리즈 버전 =====
Release : 3.10.0-957.12.1.el7.x86_64
CentOS Linux release 7.7.1908 (Core)

===== 운영 체제 이름 =====
OS name : Linux
===== 운영 체제 정보 =====
Linux version 3.10.0-957.12.1.el7.x86_64 (mockbuil
ld@kbuilder.bsys.centos.org) (gcc version 4.8.5 2
0150623 (Red Hat 4.8.5-36) (GCC) ) #1 SMP Mon Apr
29 14:59:59 UTC 2019

[khm970514@lily ch4]$
```

-a옵션

all옵션입니다. c옵션만 파일의 길이가 길기 때문에 제외하고 모두 출력해주었습니다.

-u옵션

utsname구조체의 sysname, nodename, release, version, machine을 출력해주었습니다.

-i옵션

sysinfo함수를 사용하여 시스템통계정보를 출력해주었습니다.

-f옵션

sysconf함수를 사용하여 시스템자원정보를 출력해주었습니다.

-m옵션

utsname.h헤더에 정의되어있는 utsname 구조체의 machine멤버를 출력해주었습니다.

운영체제가 동작하는 하드웨어의 표준이름(아키텍처)입니다.

-n옵션

utsname.h헤더에 정의되어있는 utsname 구조체의 nodename멤버를 출력해주었습니다.

네트워크를 통해 통신할 때 사용하는 시스템의 이름입니다.

-o옵션

운영체제에대한 정보를 출력합니다. /etc/os-release 파일을 getdelim함수를 사용하여 출력해주었습니다.

-r옵션

utsname.h헤더에 정의되어있는 utsname 구조체의 release멤버를 출력해주었습니다.

/etc/redhat-release파일을 getdelim함수를 사용하여 출력해주었습니다.

-s옵션

utsname.h헤더에 정의되어있는 utsname 구조체의 sysname멤버를 출력해주었습니다.

-v옵션

운영체제의 버전에 대한 정보를 출력합니다. /proc/version 파일을 getdelim함수를 사용하여 출력해주었습니다.

-c옵션

cpu에 대한 정보를 출력합니다. /proc/cpuinfo 파일을 getdelim함수를 사용하여 출력해주었습니다.