

4주차 1차시 시스템 관리

【학습목표】

1. 리눅스 시스템 정보를 설명할 수 있다.
2. 리눅스 시스템 관련 함수를 사용할 수 있다.

학습내용1 : 리눅스 시스템

1. 리눅스 시스템 개요

① 시스템 정보 검색 및 설정 함수
하드웨어

시스템 환경

- 운영체제 종류 관련 정보
- 메모리 페이지의 크기
- 최대 패스워드 길이 등

시스템 정보 검색 및 설정 함수

기능	함수원형
시스템 기본 정보 검색	int uname(struct utsname *name);
시스템 정보 검색과 설정	long sysinfo(int command, char *buf, long count);
시스템 자원 정보 검색	long sysconf(int name);
파일과 디렉토리 관련 자원 검색	long pathconf(const char *path, int name); long fpathconf(itn fildes, int name);

② 사용자 정보 관련 함수

사용자 계정 정보

그룹 정보

로그인 기록 정보

기능	함수원형
로그인명 검색	char *getlongin(void); char *cuserid(cahr *s);
UID 검색	uid_t getuid(void); uid_t geteuid(void);
패스워드 검색	struct passwd *getpwuid(uid_t uid); struct passwd *getpwnam(const char *name);
패스워드 파일 검색	struct passwd *getpwent(void); void setpwent(void); void endpwent(void); struct passwd *fgetpwent(FILE *fp);
새도우 정보 검색	struct spwe *getspname(const char *name);
새도우 파일 검색	struct spwd *getspent(void); void sets[ent(void); void endspent(void); struct spwd *fgetspent(FILE *fp);
그룹 정보 검색	gid_t getgid(void); gid_t getegid(void);
그룹 파일 검색	struct group *getgrnam(const char *name); struct group *getgrgid(gid_t gid); struct gorup *getgrent(void); void setgrent(void); void endgrent(void); struct group *fgetgrent(FILE *fp);
로그인 기록 검색	struct utmpx *gettutxent(void); void setutxent(void); void endutxent(void); int utmpxname(const char *file);

③ 시간 정보 검색 함수
파일 및 디렉터리 생성
사용자 정보 변경 등
시간 정보 검색 함수

기능	함수원형
현재 시간 정보 검색	time_t time(time_t *tloc); int gettimeofday(struct timeval *fp, void *tzp);
시간대 설정	void tzset(void);
시간정보 분해	struct tm *localtime(const time_t *clock); struct tm *gmtime(const time_t *clock);
초 단위 시간 생성	time_t mktime(struct tmm *timeptr);
형식 지성 시간 출력	char *ctime(const time_t *clock); char *asctime(const struct tm *tm); size_t strftime(char *restrct s, size_t maxsize, const char *restrit format, const struct tm *restrict timeptr);

학습내용2 : 리눅스 시스템 자원

1. 운영체제 기본 정보 검색

① 시스템에 설치된 운영체제에 대한 기본 정보 검색

```
# uname -a
```

```
SunOS hanbit 5.10 Generic_118855-33 i86pc i386 i86pc
운영체제명 호스트명 릴리즈 레벨 버전 번호 하드웨어 형식명 CPU명 플랫폼명
```

시스템은 인텔PC고 솔라리스 10운영체제가 설치되어 있고, 호스트명은 hanbit

② 운영체제 정보 검색 함수 : uname(2)

```
#include <sys/utsname.h>

int uname(struct utsname *name);
```

name : utsname 구조체 주조

* uname 구조체 - sys/utsname.h

```
struct utsname {
    char sysname[_SYS_NMLN];
    char nodename[_SYS_NMLN];
    char release[_SYS_NMLN];
    char version[_SYS_NMLN];
    char machine[_SYS_NMLN];
};
```

uname 구조체에 운영체제 정보 저장

sysname : 현재 운영체제 이름

nodename : 호스트명

release : 운영체제의 릴리즈 번호

version : 운영체제 버전 번호

machine : 하드웨어 아키텍처 이름

```
01 #include <sys/utsname.h>
02 #include <stdlib.h>
03 #include <stdio.h>
04
05 int main(void) {
06     struct utsname uts;
07
08     if (uname(&uts) == -1) {
09         perror("uname");
10         exit(1);
11     }
12
13     printf("OSname : %s\n", uts.sysname);
14     printf("Nodename : %s\n", uts.nodename);
15     printf("Release : %s\n", uts.release);
16     printf("Version : %s\n", uts.version);
17     printf("Machine : %s\n", uts.machine);
18
19     return 0;
20 }
```

```
# ex4_1.out
OSname : SunOS
Nodename : hanbit
Release : 5.10
Version : Generic_118855-33
Machine : i86pc
```

③ 시스템 정보 검색과 설정 : sysinfo(2)

command에 검색하거나 설정할 명령 지정

성공 시 결과값, 실패 시 -1 리턴.

```
#include <sys/systeminfo.h>

long sysinfo(int command, char *buf, long count);
```

command : 검색 또는 설정할 명령

buf : 버퍼 주소

count : 버퍼 길이, 최댓값 257

command에 사용할 상수의 범주

상수 범위	예약된 용도
1 ~ 256	<ul style="list-style-type: none"> • 유닉스 표준에서 정의한 상수 • 정보를 검색(get)하는 데 사용하기 위해 예약된 번호
257 ~ 512	<ul style="list-style-type: none"> • 유닉스 표준에서 정의한 상수 • 정보를 정의(set)하는 데 사용하기 위해 예약된 번호 • 검색 번호 + 256으로 번호 할당
513 ~ 768	<ul style="list-style-type: none"> • 솔라리스에서 추가로 정의한 상수 • 정보를 검색(get)하는 데 사용하기 위해 예약된 번호
769 ~ 1024	<ul style="list-style-type: none"> • 솔라리스에서 추가로 정의한 상수 • 정보를 정의(set)하는 데 사용하기 위해 예약된 번호 • 검색 번호 + 256으로 번호 할당

유닉스 표준에서 정의한 정보 검색용 상수

상수	설명
SI_SYSNAME(1)	운영체제명을 리턴한다. uname 함수의 sysname 항목과 같은 값이다.
SI_HOSTNAME(2)	uname 함수의 nodename 항목과 같은 값으로, 현재 시스템의 호스트명을 리턴한다.
SI_VERSION(4)	uname 함수의 version 항목과 같은 값을 리턴한다.
SI_MACHINE(5)	하드웨어 형식 값을 리턴한다. uname 함수의 machine 항목과 같은 값이다.
SI_ARCHITECTURE(6)	하드웨어의 명령어 집합 아키텍처(ISA, Instruction Set Architecture) 정보를 리턴한다. 예를 들면, sparc, mc68030, i386 등이다.
SI_HW_SERIAL(7)	하드웨어 장비의 일련번호를 리턴한다. 기본적으로 이 일련번호는 중복되지 않는다. SI_HW_PROVIDER 값과 함께 사용하면 모든 SVR4 업체의 제품을 구별하는 유일한 번호가 된다.
SI_HW_PROVIDER(8)	하드웨어 제조사 정보를 리턴한다.
SI_SRPC_DOMAIN(9)	Secure RPC(Remote Procedure Call) 도메인명을 리턴한다.

유닉스 표준에서 정의한 정보 설정용 상수

상수	설명
SI_SET_HOSTNAME(258)	호스트명을 설정한다. 이 명령은 root 사용자만 사용할 수 있다.
SI_SET_SRPC_DOMAIN(265)	Secure RPC 도메인을 설정한다.

```

01 #include <sys/systeminfo.h>
02 #include <stdlib.h>
03 #include <stdio.h>
04
05 int main(void) {
06     char buf[257];
07
08     if (sysinfo(SI_HW_SERIAL, buf, 257) == -1) {
09         perror("sysinfo");
10         exit(1);
11     }
12     printf("HW Serial : %s\n", buf);
13
14     if (sysinfo(SI_ISALIST, buf, 257) == -1) {
15         perror("sysinfo");
16         exit(1);
17     }
18     printf("ISA List : %s\n", buf);
19
20     return 0;
21 }

```

하드웨어 일련번호 검색

사용가능한 아키텍처 목록검색

```

# ex4_2.out
HW Serial : 545486663
ISA List : amd64 pentium_pro+mmx pentium_pro pentium+mmx pentium
i486 i386 i86

```


시스템 이름 변경 예

```

01 #include <sys/systeminfo.h>
02 #include <stdlib.h>
03 #include <stdio.h>
04 #include <string.h>
05
06 int main(void) {
07     char buf[257];
08
09     if (sysinfo(SI_HOSTNAME, buf, 257) == -1) {
10         perror("sysinfo");
11         exit(1);
12     }
13     printf("Before Hostname : %s\n", buf);
14
15     strcpy(buf, "hbooks");
16     if (sysinfo(SI_SET_HOSTNAME, buf, 257) == -1) {
17         perror("sysinfo");
18         exit(1);
19     }
20
21     if (sysinfo(SI_HOSTNAME, buf, 257) == -1) {
22         perror("sysinfo");
23         exit(1);
24     }
25     printf("After Hostname : %s\n", buf);
26
27     return 0;
28 }

```

호스트 이름 변경

```

# ex4_3.out
Before Hostname : hanbit
After Hostname : hbooks

```


④ 시스템 자원 정보 검색

; 하드웨어에 따라 사용할 수 있는 자원들의 최댓값 검색
; sysconf(), fpathconf(), pathconf()

* 시스템 자원 정보 검색 : sysinfo(3)

검색할 정보를 나타내는 상수를 사용해야 한다.

sys/unistd.h 파일에 정의

성공 시 시스템 자원값 또는 옵션값, 실패 시 -1 리턴

```
#include <unistd.h>

long sysconf(int name);
```

name : 검색할 정보를 나타내는 상수

```
01 #include <unistd.h>
02 #include <stdio.h>
03
04 int main(void) {
05     printf("Clock Tick : %ld\n", sysconf(_SC_CLK_TCK));
06     printf("Max Open File : %ld\n", sysconf(_SC_OPEN_MAX));
07     printf("Max Login Name Length : %ld\n", sysconf(_SC_LOGNAME_MAX));
08
09     return 0;
10 }
```

```
# ex4_4.out
Clock Tick : 100
Max Open File : 256
Max Login Name Length : 8
```

* 파일과 디렉토리 관련 자원 검색 : fpathconf(3), pathconf(3)

성공 시 경로(path)나 파일기술자에 지정된 파일에 설정된 자원값이나 옵션값을 정수로 리턴, 실패 시 -1 리턴

```
#include <unistd.h>

long pathconf(const char *path, int name);
long fpathconf(int fildes, int name);
```

path : 파일이나 디렉토리 경로

name : 검색할 정보를 지정하는 상수

fildes : 파일 기술자

name 사용할 상수

상수	설명
_PC_LINK_MAX(1)	디렉토리 혹은 파일 하나에 가능한 최대 링크 수를 나타낸다.
_PC_NAME_MAX(4)	파일명의 최대 길이를 바이트 크기로 나타낸다.
_PC_PATH_MAX(5)	경로명의 최대 길이를 바이트 크기로 나타낸다.

```
01 #include <unistd.h>
02 #include <stdio.h>
03
04 int main(void) {
05     printf("Link Max : %ld\n", pathconf(".", _PC_LINK_MAX));
06     printf("Name Max : %ld\n", pathconf(".", _PC_NAME_MAX));
07     printf("Path Max : %ld\n", pathconf(".", _PC_PATH_MAX));
08
09     return 0;
10 }
```

```
# ex4_5.out
Link Max : 32767
Name Max : 255
Path Max : 1024
```

【학습정리】

1. 시스템 정보 검색 및 설정 함수

기능	함수원형
시스템 기본 정보 검색	int uname(struct utsname *name);
시스템 정보 검색과 설정	long sysinfo(int command, char *buf, long count);
시스템 자원 정보 검색	long sysconf(int name);
파일과 디렉토리 관련 자원 검색	long pathconf(const char *path, int name); long fpathconf(itn fildes, int name);

2. 사용자 정보 관련 함수

기능	함수원형
로그인명 검색	char *getlogin(void); char *cuserid(cahr *s);
UID 검색	uid_t getuid(void); uid_t geteuid(void);
패스워드 검색	struct passwd *getpwuid(uid_t uid); struct passwd *getpwnam(const char *name);
패스워드 파일 검색	struct passwd *getpwent(void); void setpwent(void); void endpwent(void); struct passwd *fgetpwent(FILE *fp);
새도우 정보 검색	struct spwe *getspname(const char *name);
새도우 파일 검색	struct spwd *getspent(void); void sets[ent(void); void endspent(void); struct spwd *fgetspent(FILE *fp);
그룹 정보 검색	gid_t getgid(void); gid_t getegid(void);
그룹 파일 검색	struct group *getgrnam(const char *name); struct group *getgrgid(gid_t gid); struct gorup *getgrent(void); void setgrent(void); void endgrent(void); struct group *fgetgrent(FILE *fp);
로그인 기록 검색	struct utmpx *getutxent(void); void setutxent(void); void endutxent(void); int utmpxname(const char *file);

3. 시간 정보 검색 함수

기능	함수원형
현재 시간 정보 검색	time_t time(time_t *tloc); int gettimeofday(struct timeval *fp, void *tzp);
시간대 설정	void tzset(void);
시간정보 분해	struct tm *localtime(const time_t *clock); struct tm *gmtime(const time_t *clock);
초 단위 시간 생성	time_t mktime(struct tmm *timeptr);
형식 지성 시간 출력	char *ctime(const time_t *clock); char *asctime(const struct tm *tm); size_t strftime(char *restrct s, size_t maxsize, const char *restrit format, const struct tm *restrict timeptr);