**강의명: 유닉스 시스템**

**실습 번호: 5**

**실습 제목: 시스템 데이터 파일 및 정보(System Data Files and Information)**

**학생 이름: 황귀훈**

**학번: 201710885**

**1. 사용자 정보**

**1.1**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <pwd.h>**

**void myuser(const char \*user) {**

**struct passwd \*buf;**

**setpwent();**

**if ((buf=getpwnam(user)) == NULL) {**

**fprintf(stderr, "%s: unknown user\n", user);**

**exit(1);**

**}**

**endpwent();**

**printf("%s, %d, %d, %s, %s, %s\n", buf->pw\_name, buf->pw\_uid, buf->pw\_gid, buf->pw\_gecos, buf->pw\_dir, buf->pw\_shell);**

**}**

**int main(int argc, char \*argv[])**

**{**

**myuser(argv[1]);**

**return 0;**

**}**

번호1 사용자 정보는 argv[1]에 주어지는 사용자에 대하여 그 사용자의 이름, uid, gid, 사용자 정보, 홈 디렉토리, 쉘 프로그램을 순서대로 format "%s, %d, %d, %s, %s, %s\n"으로 출력하는 프로그램이다. 만약 argv[1]에 주어진 사용자가 존재하지 않으면 표준 오류로 "unknown user\n"을 출력해야 한다.

함수 시작 전 헤더파일을 include해줘야 하는데 각각의 의미는 다음과 같다. <stdio.h>는 fprintf()함수를 <stdlib.h>는 exit()함수를 사용하기 위해 include하였고, struct passwd를 사용하기 위해 <pwd.h>를 include했다.

이 프로그램에 목적에 맞게 주어지는 user name에 대한 정보를 알아내기 위해 myuser()함수를 작성하였다. Myuser()함수에서는 먼저 passwd구조체의 주소를 정의하고 getpwnam()함수를 사용하여 myuser()함수로 넘어온 user의 정보를 passwd구조체에 담았다. getpwnam()함수의 사용 전에 setpwent()함수를 사용하여 read offset을 처음으로 옮기고 getpwnam()함수의 사용 후로 endpwent()함수를 사용해 read를 close해 주었다. 이 때 만약 getpwnam()의 반환 값으로 NULL을 반환했다면 실패를 한 경우 이므로 에러 메시지를 출력하고 exit(1)함수로 종료하도록 하였다.

getpwnam()함수가 성공했을 경우 문제의 format에 맞게 주어진 user의 사용자의 이름, uid, gid, 사용자 정보, 홈 디렉토리, 쉘 프로그램을 출력하였다.

Main()함수는 위에 작성한 myuser()함수에 인수로 argv[1]을 넘겨주고 종료하도록 간단하게 작성하였다.

**1.2**

****

**1.3**

****

**2. 시스템 정보**

**2.1**

**#include <stdio.h>**

**#include <stdlib.h>**

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

**void mysystem(void) {**

**struct utsname \*buf = (struct utsname \*)malloc(sizeof(struct utsname));**

**if (uname(buf) == -1) {**

**perror("uname");**

**exit(1);**

**}**

**fprintf(stdout, "%s\n%s\n%s\n%s\n%s\n", buf->sysname, buf->nodename, buf->release, buf->version, buf->machine);**

**free(buf);**

**}**

**int main(int argc, char \*argv[])**

**{**

**mysystem();**

**return 0;**

**}**

번호2 시스템 정보는 현재 사용 중인 유닉스 시스템의 정보를 출력하는 프로그램으로, 이 프로그램은 커널 이름,노드 이름, 커널 release, 커널 버전, 하드웨어 타입을 형식 "%s, %s, %s, %s, %s\n" 으로 출력해야 한다.

함수 시작 전 헤더파일을 include해줘야 하는데 각각의 의미는 다음과 같다. <stdio.h>는 fprintf()함수를 <stdlib.h>는 perror(), exit()함수를 사용하기 위해 include하였고, struct utsname을 사용하기 위해 <sys/utsname.h>을 include했다.

이 프로그램의 목적에 맞게 시스템 정보를 알아내기 위해 mysystem()함수를 작성하였다. mysystem()에서 시스템의 정보를 담는 구조체인 utsname을 선언한 뒤 uname()함수를 사용하여 시스템 정보를 utsname구조체에 담았다. 이 때 struct utsname \*형식의 buf를 uname함수의 인수로 전달하기 위해 malloc()함수를 사용하여 메모리를 할당 해 주었고, 함수의 수행이 끝난 후 free()함수를 사용하여 메모리를 할당을 제거해 주었다.

만약 uname()함수의 반환 값으로 -1이 나온다면 오류가 발생한 것이기 때문에 perror(“uname”)함수를 사용하여 에러메시지를 출력하고 exit(1)함수를 사용하여 종료하였다. uname()함수가 잘 수행 되면 utsname구조체에 시스템 정보가 담길 것이고 fprintf()함수를 사용하여 문제에 주어진 format에 맞게 커널 이름, 노드 이름, 커널 release, 커널 버전, 하드웨어 타입을 출력하도록 하였다.

main()함수는 단순하게 mysystem()함수를 호출한 후 종료한다.

**2.2**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**2.3**

****

**3. 시간 정보**

**3.1**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <time.h>**

**void mydate(void) {**

**time\_t \*t = (time\_t \*)malloc(sizeof(time\_t));**

**char \*str;**

**if (time(t) == -1) {**

**perror("time");**

**exit(1);**

**}**

**printf("%lu\n", \*t);**

**if ((str = ctime(t)) == NULL) {**

**perror("ctime");**

**exit(1);**

**}**

**printf("%s",str);**

**free(t);**

**}**

**int main(void)**

**{**

**mydate();**

**return 0;**

**}**

번호3 시간 정보는 현재 시간을 화면에 출력하는 프로그램으로, 출력의 첫 줄에는 현재 calendar time을 형식 "%lu\n"으로 출력하고, 출력의 둘째 줄에는 현재 string time을 형식 "%s\n"으로 출력해야 한다.

함수 시작 전 헤더파일을 include해줘야 하는데 각각의 의미는 다음과 같다. <stdio.h>는 fprintf()함수를 <stdlib.h>는 perror(), exit()함수를 사용하기 위해 include하였고, 시간과 날짜를 표현하기 위하여 사용하는 다양한 함수들을 사용하기 위해<time.h>를 include했다.

이 프로그램의 목적에 맞게 시간을 여러 형식으로 출력하기 위해 mydate()함수를 작성하였다. mydate()함수에서는 time() 함수를 사용하여 calendar time을 알아낸 뒤에 이를 “%lu\n”형식으로 출력하였다. 이 때 time\_t \* 형식의 변수 t를 time()함수의 인수로 전달하기 위해 malloc()함수를 사용하여 메모리를 할당했고, 사용을 마친 뒤 free()함수를 사용하여 메모리를 해제했다. 그 다음에 위에서 알아낸 calendar time을 ctime()함수의 인수로 전달해 string time을 알아냈고, 이를 “%s”형식으로 출력하였다.

각 함수 time(), ctime()은 오류 시 NULL을 반환하기 때문에 if문을 사용하여 오류 발생 시 perror()함수를 사용하여 에러 메시지를 출력한 후 exit(1)함수를 사용해 프로그램을 종료했다.

Main()함수는 위에서 작성한 mydate()함수를 호출한 뒤 종료하도록 작성하였다.

**3.2**

텍스트이(가) 표시된 사진

자동 생성된 설명

끝.