

1. [실습 5:cmd-1]의 5번 문제인 whoami 명령어 구현 시, getlogin() 함수 대신 getpwuid() 함수를 사용하도록 프로그램을 수정하라. 강의노트 6장 pp.5~10, 교재 pp.194~200을 참고하라.

- 1) 찾기 기능으로 기존의 whoami() 함수를 찾아가, 함수 안의 기존 코드를 주석 처리하라. 대신 현재 cmd 프로그램을 실행시킨 사용자의 계정정보를 직접 구해 와 계정이름을 출력하게 할 것이다.
- 2) 먼저 계정정보 구조체의 포인터 변수인 pwp를 선언한다.
struct passwd *pwp;
- 3) 사용자의 계정정보를 얻어 오기 위해선 getpwuid() 함수(함수2)를 호출해야 하는데, 이 함수를 호출하기 위해선 함수 호출 시 인자로 사용자 id 값을 주어야 한다. 현재 cmd 프로그램을 실행시킨 사용자의 id는 어떻게 구할까? getuid() 함수(함수1)의 리턴 값이다. (강의노트 8장 p.3 참조)
- 4) 이제 getpwuid() 함수를 호출하여 리턴 값을 포인터 변수 pwp에 저장하라. 이 함수 호출 시 인자로 getuid()의 리턴 값을 직접 준다. 이제 pwp는 사용자 계정 정보를 담고 있는 passwd 구조체를 포인터 하고 있다.
- 5) 위 3)과 4)의 코드를 한 문장으로 줄이면, pwp = 함수2호출 (함수1호출) 형태의 구조인데, 함수1을 먼저 호출하고 리턴 값을 바로 함수2의 인자로 전해주는 것이다.
- 6) 이제 pwp가 포인터하는 구조체의 멤버 중 사용자 계정 이름을 가진 멤버를 문자열로 출력하라. (멤버 이름은 강의노트 참조)
- 7) 두 함수를 \$ man으로 찾아 헤드 파일을 include시킨다.
getuid() 함수의 리턴 데이터 타입 uid_t는 어디에 선언되어 있을까? 당연히 getuid()를 선언한 헤더파일에 있음.
- 8) make한 후 cmd를 실행시켜 whoami를 테스트하라.

2. [실습 6:cmd-2]의 1번 문제인 cd 명령어 구현 시, cd 다음에 [디렉토리 이름]을 주지 않았을 경우, 자신의 홈 디렉토리를 직접 argv[0]에 설정해 주었다. 이전 시스템 계정에 등록된 자신의 홈 디렉토리를 argv[0]에 지정해 주도록 프로그램을 수정하라.

- 1) cd() 함수의 if (argc가 0)이면 {
// 문장이 여러 개이므로 {}해야 함
- i) 위 실습 1.2)처럼 pwp 포인터 변수를 선언한다.
- ii) 위 실습 1.5)처럼 getpwuid()를 호출한 후 리턴 값을 pwp에 저장한다. (함수 인자도 동일하게 지정)
- iii) 이제 pwp가 포인터하는 구조체 멤버 중 사용자 홈 디렉토리를 가진 멤버 값을 argv[0]에 설정한다. (멤버 이름은 강의노트 참조) **chdir(argv[0])**

iv) 위 if 문장의 { 의 짝인 }를 입력한다.

- 2) make한 후 cmd를 실행시켜 다른 디렉토리로 이동한 후 cd[엔터]를 하라. 자신의 홈으로 이동하는가?

3. id 명령어를 구현하라.

- 1) 먼저 id() 함수를 정의하자.

```
// 사용자ID(계정이름)와 groupID(그룹명)를 보여 주는 명령어
// 사용법: id [계정이름]
// 다른 사람 계정 이름을 지정해도 됨;
// []는 주지 않아도 되며, 이 경우 본인 정보 출력
// argv[0] -> "계정이름"; [계정이름]을 준 경우
// argc = [계정이름]을 준 경우 1, 안 준 경우 0
void id(void) {} // 기존 함수 형태로 할 것
```

- 2) 강의노트 6장 pp.5~10, 교재 pp.194~200을 참조하여 id() 함수를 구현하자.

i) 다음 두 변수를 선언한다.

```
struct passwd *pwp;
struct group *grp;
```

ii) 명령어 인자로 계정 이름을 주었으면 getpwnam() 함수(함수1)를 호출하고, 주지 않았으면 getpwuid() 함수(함수2)를 호출한다. 두 함수는 사용자 정보를 담고 있는 구조체에 대한 포인터를 리턴 한다. 각 함수의 리턴 값을 pwp에 저장하라. getpwnam()은 인자로 id 명령어 인자로 제공된 [계정이름]을 주면 되고, getpwuid()는 uid 값을 주어야 한다. 이 uid는 getuid() 함수(함수4)를 호출한 리턴 값으로 구할 수 있다. (현 cmd를 실행시키는 사용자의 user id 값).

iii) 만약 위 두 함수의 리턴 값이 NULL이라면 아래의 에러 메시지를 출력하고 리턴 한다. 이때 매크로 함수 PRINT_ERR_RET()은 여기서 사용할 수 없다. 따라서 여기선 직접 다음 같이 출력해야 한다.

id: 잘못된 사용자 이름: "jxshim"

jxshim은 id 명령어 인자이다. 위치럼 "?"가 출력되게 하려면 printf()에서 어떻게 해야 할까?

iv) 정상적으로 사용자의 계정정보 pwp가 구해졌다면, 이제 사용자의 그룹 정보를 얻어오는 getgrgid() 함수(함수3)를 호출하고, 리턴 값을 grp에 저장하라. 만약 이 함수의 리턴 값이 에러라면 iii) 같은 에러를 출력하고 리턴 한다. 이 함수의 인자로 group id를 주어야 하는데, 이는 pwp 구조체의 멤버를 활용하면 된다. 따라서 pwp가 NULL인데도 이 함수를 호출하면 프로

```
jxshim:[~] $ id
uid=1001(jxshim) gid=1001(jxshim)
jxshim:[~] $ id jxshim1
uid=1002(jxshim1) gid=1002(jxshim1)
jxshim:[~] $
```

그램 죽으니 NULL이면 호출하지 않도록 유의하라.

- v) 이제 pwp와 grp가 포인터하는 구조체 멤버를 찾아서 출력해 주면 된다. 출력양식은 기존 명령 창에서 id 명령어를 실행해서 나오는 결과를 참조하라.

- v) 여러분은 이렇게 구현할 것이다.

```

if (계정이름을 주었다면)
    pwp = 함수1호출(사용자계정이름);
else
    pwp = 함수2호출(함수4호출());
if (pwp == NULL) { // 에러이면
    에러 출력;
    return;
}
grp = 함수3호출
if (grp == NULL) { // 에러이면
    에러 출력;
    return;
}
pwp와 grp 멤버 출력
-----
그러나 아래처럼 한 문장으로 줄이면 더 효율적이다.
-----
pwp = (계정이름 주었다면)? 함수1호출: 함수2호출;
if ( (pwp == NULL) ||
      ((grp = 함수3호출) == NULL) )
    에러 출력;
else
    pwp와 grp 멤버 출력
-----

```

위 경우 pwp가 NULL이면 함수3을 호출하지 않는다.

- cmd_tbl[] 배열에 id 관련 배열 원소를 추가하라. 위 1)번의 주석문 중 사용법을 참조하기 바란다.
- 위 네 함수의 헤드 파일을 \$ man으로 찾아 include시킨다.
- make하라. 그런 후 cmd를 실행시키고 명령어로 다음을 테스트하라. "> "는 cmd의 명령 프롬프트다.

```

> id           // 본인의 계정정보
> id 친구계정  // 친구의 계정정보
> id hoha      // 없는 계정; 아래 에러 출력
id: 잘못된 사용자 이름: "hoha"

```

4. date 명령어를 구현해 보자.

- 1) 먼저 date() 함수를 정의하자.

```

// 오늘 현재의 "년월일시분초"를 보여 주는 명령어
// 사용법: date
-----

```

- 2) 강의노트 6장 pp.17~24, 교재 pp.207~211을 참조하여 date() 함수를 구현하자.

- i) 먼저 date() 내에서 다음 세 변수를 선언한다.

```

char    stm[128];
a) ttm 변수를 선언하되, 초기 값으로 time(NULL) 함수를 호출하여 리턴 값을 저장하라. (calendar time 구함) 변수 타입은 함수의 리턴 타입을 참조하라.
b) ltm 변수를 선언하되, 초기 값으로 localtime()을 호출하여 리턴 값을 저장하라. (한국시간 broken down time 구함) 함수의 인자는 앞의 ttm을 활용한다. 함수 인자의 데이터 타입을 잘 보고 지정하라.
-----

```

- ii) 이제 한국시간인 ltm을 년월일시분초 문자열로 자동 변환하는 asctime()을 호출하고, 리턴된 문자열 주소를 printf()로 바로 출력하라. "atm: 시간정보" 형태로 출력하라. 하나의 printf로 이 모두를 처리하라.

- ii) Calendar time인 ttm을 바로 한국시간으로 년월일시분초 문자열로 변환하는 ctime()을 호출하고, 리턴된 문자열 주소를 printf()로 바로 출력하라. 출력 문장의 시작은 "ctm:" 형태로 하고, 하나의 printf로 모두를 처리하라.

- iii) strftime() 함수를 이용하여 여러분이 직접 위 두 출력 형태와 똑 같이 만들어 stm 배열에 저장하라. 출력 문장의 시작은 "stm:"으로 시작하게 하라. 이 함수 사용방법은 강의노트 예제를 참조하되, 변수이름이나 "%"안의 양식은 적절히 변경하라.

- iv) stm[]에 저장된 것은 문자열이다. 이를 출력하라. 하나의 문자열을 출력하는 아래의 형태의 문장은

```
printf("%s\n", 문자열변수);
```

아래 문장으로 대신하면 더 편하다.

```
puts(문자열변수);
```

- 3) cmd_tbl[] 배열에 date 관련 배열 원소를 추가하라. 위 1)번의 주석문 중 사용법을 참조하기 바란다.

- 4) 위 네 함수의 헤드 파일을 include시킨다.

- 5) make하라. 그런 후 cmd를 실행시키고 date 명령어를 테스트하라.

5. 명령 창에서 다음을 실행하여 정상임을 확인하라.

```

$ eshw 8
$ progtest 8

```

```

</home/jhshim1/up/cmd> 1: date
atm: Tue May 19 16:21:42 2020
ctm: Tue May 19 16:21:42 2020
stm: Tue May 19 16:21:42 2020

```