리눅스

사용자 관리 및 허가권 관리

개요

- 리눅스는 다중 사용자 시스템으로서, 여러 사람이 한 컴퓨터에 로그인을 하여 컴퓨터를 공유하여 사용할 수 있다.
- 따라서, 여러 사용자가 컴퓨터에 있는 파일에 대해 접근이 가능하다.

- 리눅스는 파일의 소유자가 파일에 대한 읽고, 쓰고, 실행하기에 대한 권한을 설정하여, 다른 사용자의 접근을 허가 또는 불허하는 접근 통제 메커니즘을 제공 (DAC : Discretionary Access Control)

© System Software & Security Lab@Myongji Univ.

리눅스 사용자 유형

- 2가지 사용자 유형
  - root 사용자 (super-user, administrator)
  - 일반 사용자
- root 사용자
  - 일반적으로 시스템을 설치할 때 자동으로 만들어짐
  - 로그인할 때 root로 로그인 함
- 일반 사용자
  - 일반 사용자는 수퍼유저가 사용자 계정을 생성하여 만든다.

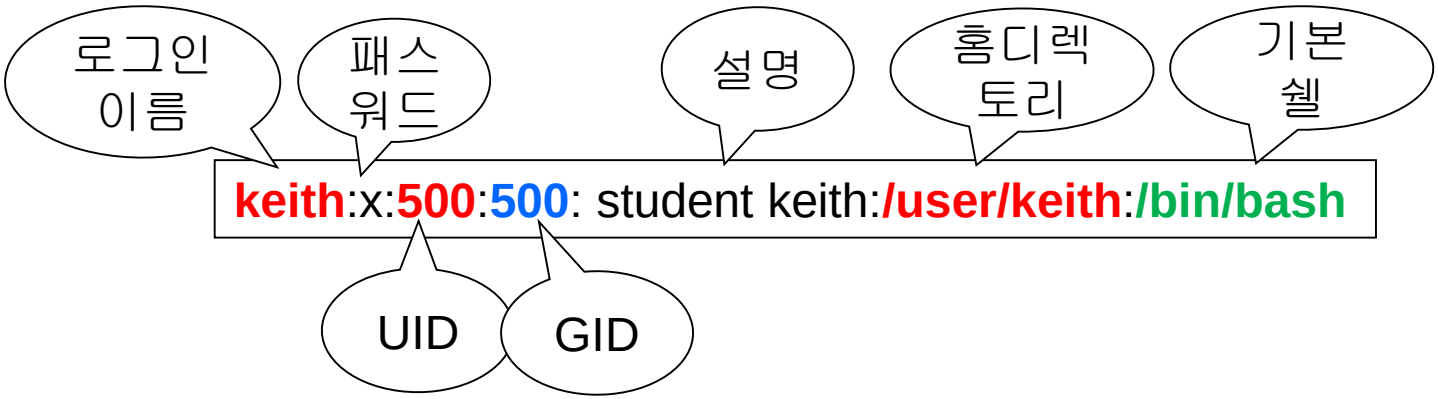
© System Software & Security Lab@Myongji Univ.

사용자 계정

- 사용자 계정(user account)
  - 사용자에게 부여되는 로그인 이름과 패스워드(password)

- 사용자 계정 정보가 관리되는 시스템 파일
  - /etc/passwd : 한 줄에 한 사용자의 정보가 있음
  - /etc/shadow : 사용자의 패스워드 정보가 있음

/etc/passwd 내용 예 : 사용자 keith의 정보



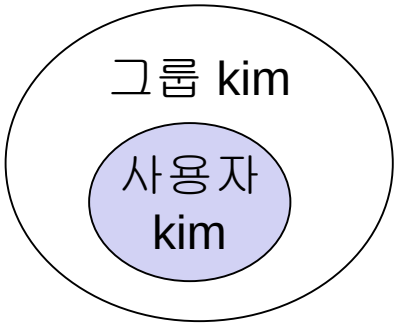
© System Software & Security Lab@Myongji Univ.

그룹(group)

- 사용자 그룹이란?
  - 사용자들을 그룹으로 묶음 (예:학생 그룹, 교수 그룹 등)
  - 같은 그룹에 있는 사용자는 같은 특징을 가진 사용자들

- 사용자는 하나 이상의 그룹에 속하게 된다.
  - 사용자 계정을 생성할 때, 자동으로 계정명과 동일한 이름의 그룹이 생성되며 그 그룹에 소속됨

    예) kim이라는 사용자 계정을 새로 만들면, 그룹 kim이 생성되고 사용자 kim은 그룹 kim에 소속됨



© System Software & Security Lab@Myongji Univ.

UID, GID

- 모든 사용자는 두 개의 식별번호(ID)를 갖는다
  - 사용자 식별번호 (UID : User ID)
  - 그룹 식별번호 (GID : Group ID)

/etc/passwd 파일의 내용

keith:x:**500**:**500**: :/user/keith:/bin/bash

UID

GID

- 그룹 정보를 관리하는 시스템 파일 : /etc/group

그룹 이름

패스워드

keith에 속한 사용자 목록

keith:x:**500**:**kim,kdhong**

GID

© System Software & Security Lab@Myongji Univ.

실습: 계정 생성하기

1. 사용자 ysryu를 생성하기

useradd ysryu

- 사용자 ysryu의 UID와 GID를 적으시오.
- /etc/group 파일에서 그룹 ysryu의 GID를 찾아 적으시오.

2. 사용자 ysryu를 생성할 때 옵션 사용하는 예

useradd –c “professor” –d /home/ysryu ysryu

useradd –g prof ysryu

useradd –s /bin/bash ysryu

© System Software & Security Lab@Myongji Univ.

실습: 패스워드 변경하기

- passwd [option] [user-name]

  passwd ysryu ..... 사용자 ysryu의 패스워드 변경

  passwd –l ysryu ..... 사용자 ysryu 계정을 임시 중단

  passwd –u ysryu ..... 사용자 ysryu 계정을 다시 허용

  passwd –d ysryu ..... 사용자 ysryu의 패스워드를 삭제

© System Software & Security Lab@Myongji Univ.

실습: 그룹 생성하기

- groupadd [option] group-name
  - -g GID : 지정하는 GID로 그룹 생성
  - -o : 지정한 GID 값이 존재하더라고 중복을 허용
  - -r : 시스템 관리용 그룹 생성

```
groupadd  linux                  // linux 그룹을 생성. GID는 자동부여

groupadd  –g 1005  linux1        // linux1 그룹을 생성. GID는 1005번으로

groupadd  -g 1005  –o  linux2    // linux2 그룹을 생성. GID 1005번이 이미
                                    있어도 생성함

groupadd  –r  sysadmin
```


© System Software & Security Lab@Myongji Univ.

실습: 사용자의 그룹 변경하기

- 사용자 계정 정보 변경하기

  usermod [option] user-name

- 사용자 그룹을 바꾸려면
  - usermod –g *GID username*

- 사용자를 다른 그룹에 추가로 속하게 하려면
  - usermod -G *GID username*

※ 그룹이 아직 존재하지 않는 다면,
groupadd 명령어로 그룹을 새로 만들어야 함

© System Software & Security Lab@Myongji Univ.

기타 명령어

- groups [username]: 사용자가 소속된 모든 그룹을 출력
  - 앞에서 만든 ysryu의 그룹을 출력하시오.

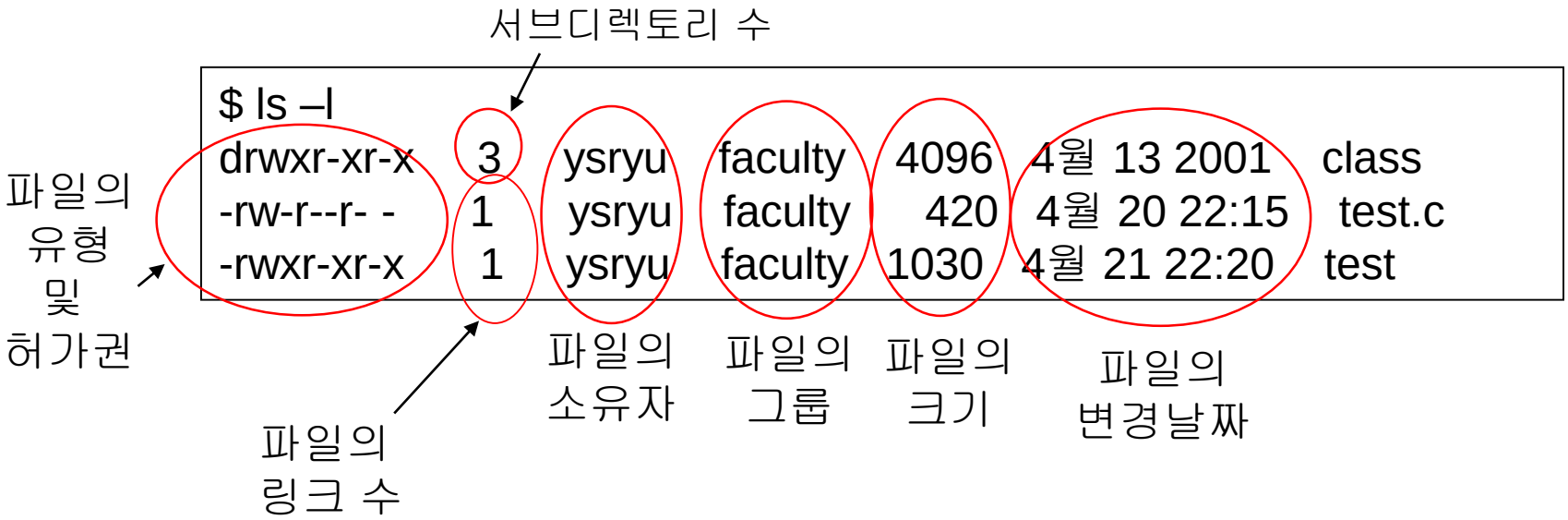
- id [username] : 사용자의 UID, GID를 출력
  - 앞에서 만든 ysryu의 UID, GID를 출력하시오.

© System Software & Security Lab@Myongji Univ.

파일의 소유권

- 파일의 소유권 (ownership)
  - 파일을 만들면, 소유자와 그룹이 부여된다.
  - 파일을 만든 사용자가 소유권을 가짐 (소유자)
  - 파일의 그룹은 소유자의 그룹으로 부여됨

    → 즉, 파일을 만든 사람의 uid와 gid로 셋팅됨



- 파일의 소유권은 수퍼 사용자 또는 소유자가 변경 가능 (chown 명령어)

© System Software & Security Lab@Myongji Univ.

파일의 사용자

- 파일 사용자의 세가지 유형
  - 파일의 소유자 (owner, user : u)
  - 파일의 그룹에 속하는 사용자 (group : g)
  - 그외 사용자 (others : o)

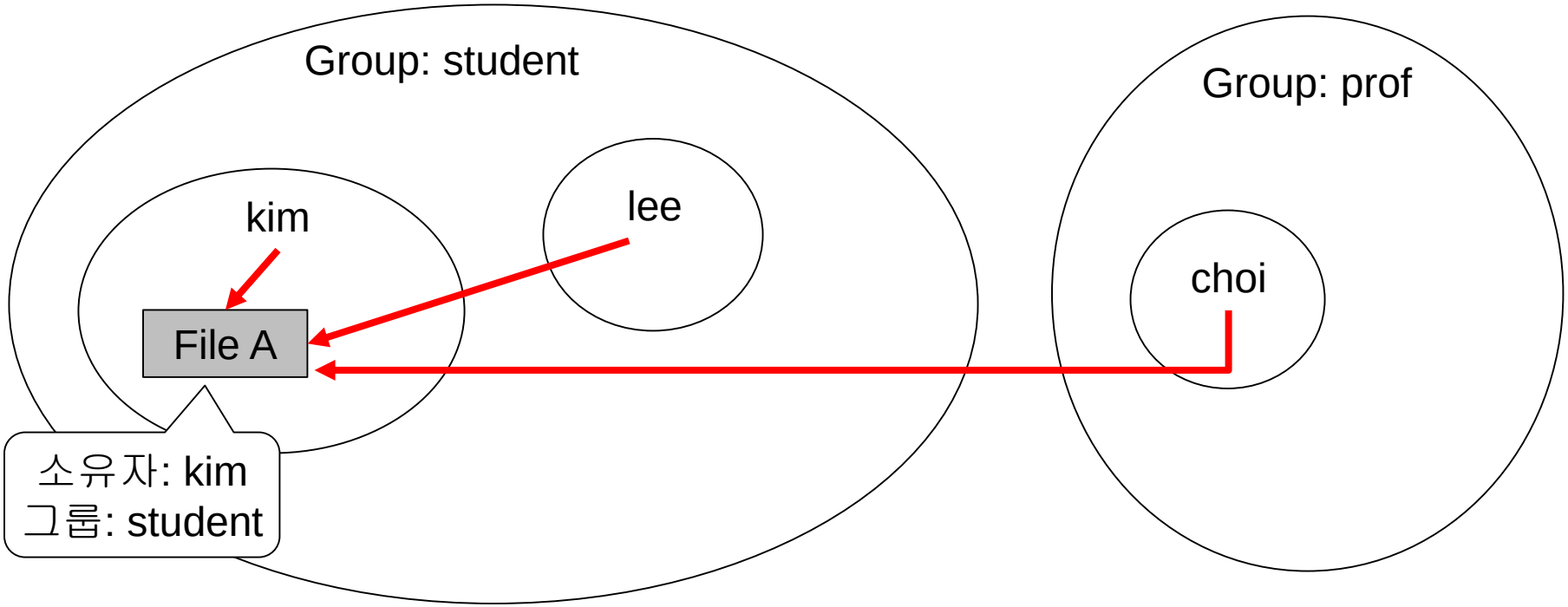
- 파일의 소유자는 사용자 유형별로 파일의 사용 권한을 다르게 할 수 있다 (chmod 명령어)
  - 예: 소유자는 read/write가 가능하게
    그룹의 사용자는 read만 가능하게
    그외 사용자는 read만 가능하게

파일의 사용(use)이란?
- 읽기(read), 쓰기(write), 실행하기(execute)
- 파일의 접근(access)이라고도 함

© System Software & Security Lab@Myongji Univ.

파일의 사용자

- 파일의 사용자 3가지 유형



© System Software & Security Lab@Myongji Univ.

파일의 허가권(permission)

- 파일의 허가권
  - 파일에 대한 접근 권한

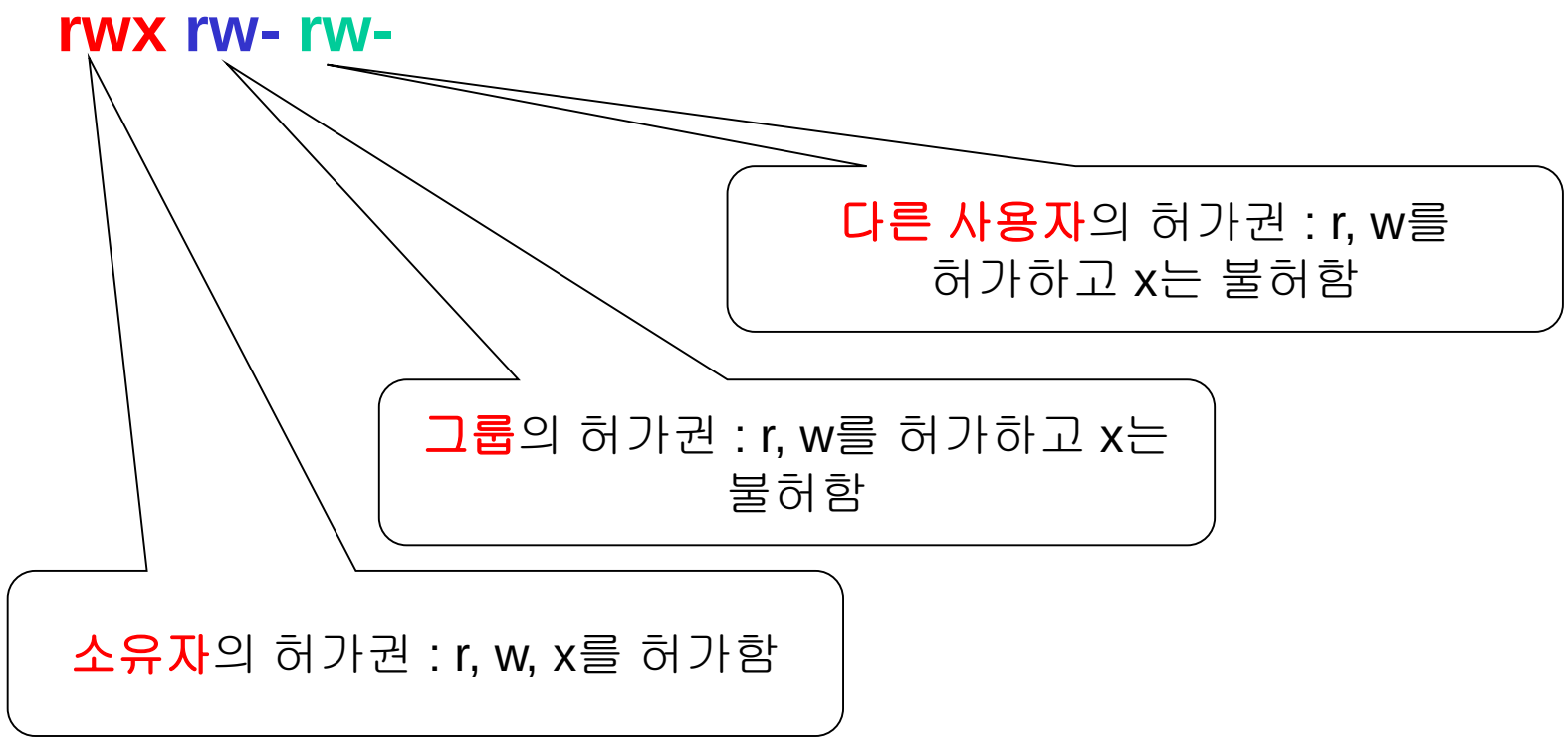
- 세가지 유형
  - 읽기 (read : r)
  - 쓰기 (write : w)
  - 수행하기 (execute :x) (디렉토리의 경우는 cd를 사용하여 디렉토리 내부로 들어가기)

- 파일의 허가권은 사용자 유형과 파일허가 유형의 조합으로 셋팅된다
  - 예) 소유자에게 읽기,쓰기를 허가하고 수행하기는 불허
  - 예) 그룹 사용자에게 읽기를 허가하고 쓰기, 수행하기는 불허
  - 예) 그외 사용자에게 읽기를 허가하고 쓰기, 수행하기는 불허

© System Software & Security Lab@Myongji Univ.

파일의 허가권

- ls –l에서 출력된 허가권의 의미



© System Software & Security Lab@Myongji Univ.

파일의 허가권 변경

- chmod 명령어
  - 파일의 소유자 (또는 root) 가 허가권을 변경

예)

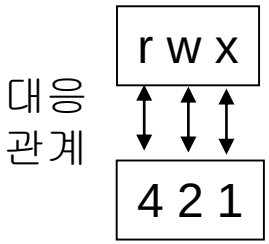
```
chmod  g+w   sample.txt
chmod  g-w,o-w    sample.txt
chmod  a+w   sample.txt            a : all (모든 사용자)
chmod  o=r    sample.txt

chmod  744   sample.txt
```


© System Software & Security Lab@Myongji Univ.

파일의 허가권

- 허가권을 지정하는 팔진수 값
  - 8진수 세자리 : 각각 소유자, 그룹, 다른사용자의 허가권을 의미
    예) 744 : 소유자 7, 그룹 4, 다른사용자 4



예) 400 : 소유자에게 읽기 허용
040 : 그룹에게 읽기 허용
004 : 다른 사용자에게 읽기 허용

소유자에게 읽기, 쓰기, 수행을 허용하려면
400 + 200 + 100 = 700

그룹에게 읽기, 수행을 허용하려면
040 + 010 = 050

위 둘을 모두 허용하려면
700 + 050 = 750

© System Software & Security Lab@Myongji Univ.

파일의 기본 허가권 : umask

- 파일 (또는 디렉토리)을 생성할 때 설정되는 기본 허가권은 umask 값에 의해 결정됨
- umask는 일반적으로 /etc/profile 또는 /etc/bashrc 등에 설정되어 있으며, 개별 사용자가 .bashrc 파일에서 설정할 수도 있음
- 파일이 생성될 때, 최고 권한에서 umask 값을 빼서 기본 권한을 정함
  - 파일의 최고 권한은 666, 디렉토리의 최고 권한은 777
- 예) umask 값이 002로 설정되어 있는 경우 (→명령어 umask 로 값 확인)
  - 파일: 666 – 002 = 664
  - 디렉토리: 777 – 002 = 775

© System Software & Security Lab@Myongji Univ.

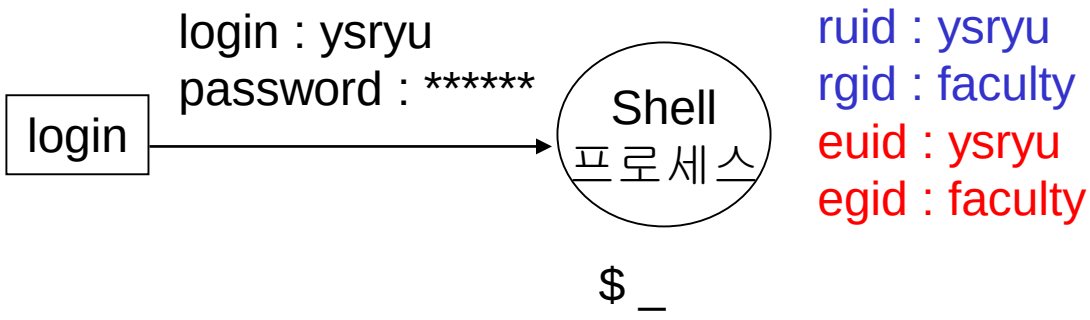
허가권의 적용 규칙

- 프로세스마다 부여받는 식별번호 4개가 있음
  - ruid : real user id
  - rgid : real group id
  - euid : effective user id
  - egid : effective group id

  } 허가권을 검사할 때 사용

프로세스(process) : 실행 중인 프로그램

- 쉘 프로세스는 ruid와 euid 값이 로그인한 사용자의 UID를 부여받음.
- rgid와 egid는 로그인한 사용자의 GID를 부여받음

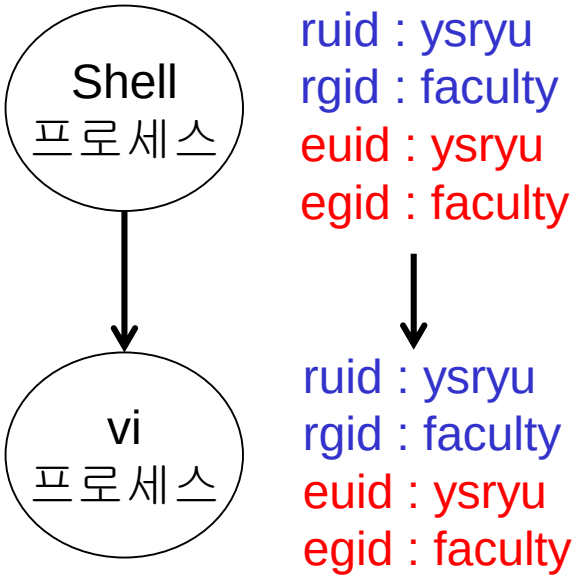


© System Software & Security Lab@Myongji Univ.

허가권의 적용 규칙

- 쉘이 수행한 자식 프로세스는 부모 프로세스인 쉘 프로세스의 4개 식별번호를 상속받는다.

$ vi /etc/passwd



© System Software & Security Lab@Myongji Univ.

허가권의 적용 규칙

- 프로세스가 어떤 파일을 사용하려고 할 때, 파일의 허가권의 적용 규칙
  - 프로세스의 euid가 파일의 소유자(uid)와 같다면, 소유자의 허가권이 적용된다.
  - 프로세스의 euid가 파일의 소유자와 다르지만, egid가 파일의 그룹 id와 같다면, 그룹의 허가권이 적용된다.
  - 프로세스의 euid와 egid가 파일의 소유자나 그룹과 다르다면, 기타 사용자의 허가권이 적용된다.

© System Software & Security Lab@Myongji Univ.

실습: 허가권의 적용 규칙

- 예: /etc/passwd 파일의 허가권을 조사하시오.
  - 소유자와 그룹은 누구인가?
  - 소유자는 이 파일에 대해 어떤 사용이 가능한가?
  - 기타 사용자는 이 파일을 vi로 읽어볼 수 있는가?
  - 기타 사용자는 이 파일을 vi로 변경할 수 있는가?

© System Software & Security Lab@Myongji Univ.

실습: 파일의 허가권

- 예: /usr/bin/passwd 파일의 허가권을 조사하시오.
  - 소유자와 그룹은 누구이고 설정된 허가권은 무엇인가?
  - 기타 사용자는 이 파일을 실행하여 암호를 변경할 수 있는가?

참고: passwd 프로그램은 암호를 변경할때
/etc/passwd 파일과 /etc/shadow 파일을 변경한다.

© System Software & Security Lab@Myongji Univ.

실행파일의 특수 허가권

- 실행 파일에 대한 특별한 허가권
  - 4000 (S_ISUID) : setuid, 소유자 허가권에서 x 대신 s로 표시됨
  - 2000 (S_ISGID) : setgid, 그룹 허가권에서 x 대신 s로 표시됨
  - 1000 (S_ISVTX) : sticky bit, 기타 허가권에서 x 대신 t, T로 표시됨

  - **setuid** : 파일을 실행하여 생성된 프로세스의 euid를 파일 소유자의 user-id로 셋팅한다.
    - 예) passwd 프로그램

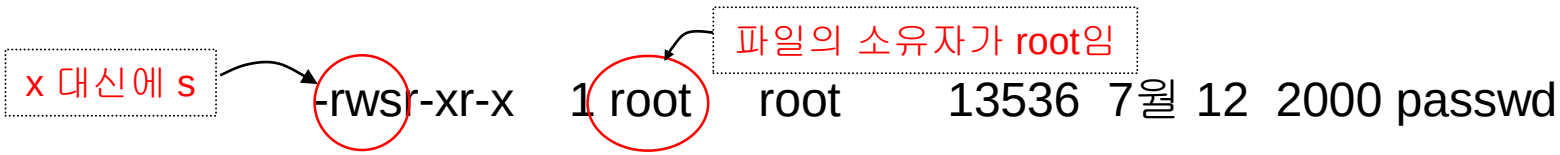
  - **setgid** : 파일을 실행하여 생성된 프로세스의 egid를 파일 그룹의 id로 셋팅한다

© System Software & Security Lab@Myongji Univ.

setuid 예

- setuid 예: passwd 프로그램

```
$ ls –l /usr/bin/passwd
```



passwd 프로그램은 사용자의 암호를 바꾸기 위해 /etc/passwd 파일을 변경한다.
그런데, 이 파일의 허가권을 보면,

```
$ ls –l /etc/passwd
```

```
-rw-r--r--  1 root  root  1739 5월 14 21:01 passwd
```

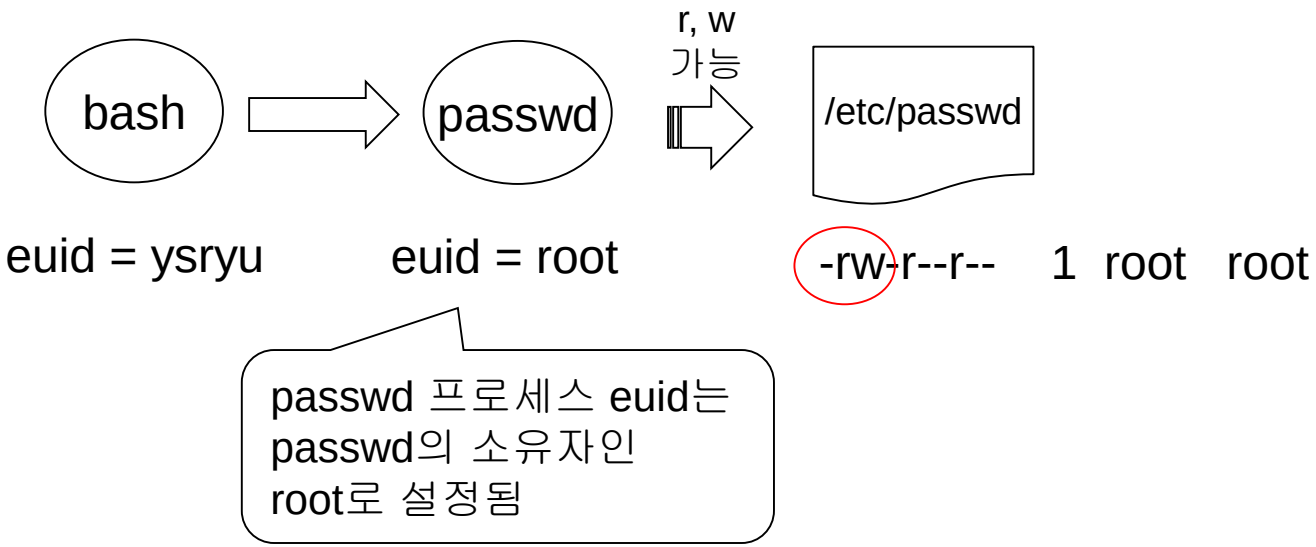
파일의 소유자는 root이고 root만이 쓰기 권한이 있다.
따라서, /usr/bin/passwd 프로그램을 수행했을 때 그 프로세스의 euid가 root이어야 함.
setuid가 설정되어 있으므로 euid가 파일의 소유자인 root로 바뀐다.

© System Software & Security Lab@Myongji Univ.

setuid 예

- 사용자 ysryu 가 패스워드를 변경하는 경우

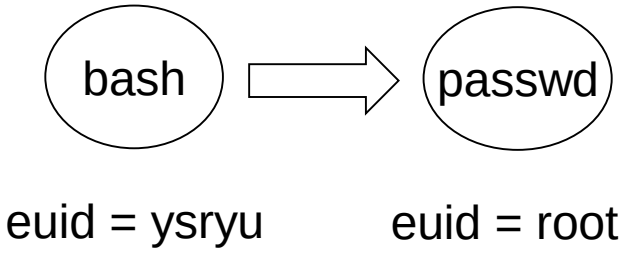
쉘 프롬프트에서 passwd 를 실행 (☞ /usr/bin/passwd 프로그램)



© System Software & Security Lab@Myongji Univ.

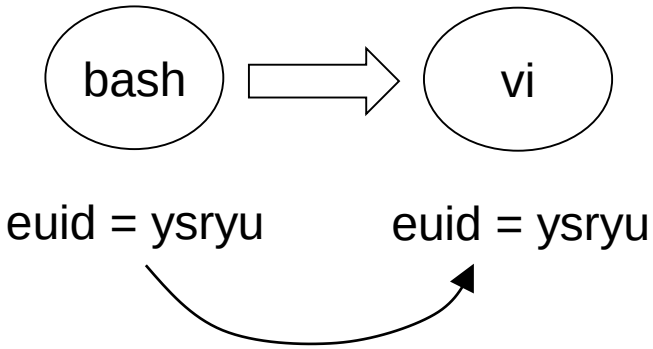
setuid 정리

- setuid가 설정되어 있는 파일(passwd)을 실행한 경우



passwd 프로세스 euid는
passwd의 소유자인
root로 설정됨

- setuid가 설정되어 있지 않은 파일(vi)을 실행한 경우



vi 프로세스의 euid는
bash 프로세스의
euid인 ysryu를 상속받음

© System Software & Security Lab@Myongji Univ.

실습 : 실행파일의 특수 허가권

- 임시로 빈 파일 생성하기

  touch sample

- setuid 설정하기

  chmod u+s sample

- setgid 설정하기

  chmod g+s sample

© System Software & Security Lab@Myongji Univ.