인덕대학교

운영체제 보안
- 프로세스 제한2

컴퓨터소프트웨어학과

김병국 교수

학습목표

- SSH 서버를 구축할 수 있다.
- 계정 별 프로세스의 기능을 제한할 수 있다.

목차

- 실습 환경 구축
- 프로세스 제한 실습

1. 환경구축 (1/2)

❑ 로그인 환경 구축

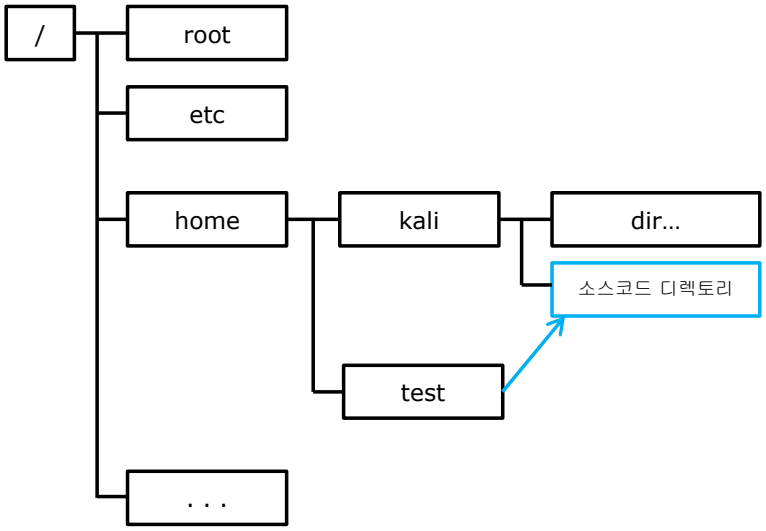
- SSH 서버
  - 대부분의 유닉스에서는 SSH(Secure Shell)를 지원함
  - 데비안 계열의 리눅스 배포판의 경우 APT(Advanced Package Tool)를 통해 관련 프로그램에 대한 설치 및 삭제 가능
  - 설치 확인
    - 명령: apt list ssh

- SSH 서버 구동
  - 명령: sudo service ssh restart
  - 재부팅 후 자동 실행:
    sudo systemctl enable ssh

1. 환경구축 (2/2)

❑ 파일 접근 권한 및 링크

- 새로 추가된 계정(이름: test)의 접근을 허용하기 위한 선행작업 추천
- /home/kali 하단의 작업중인 코드에 대한 test계정에서 쉬운 접근용 링크를 생성



2. 제한 1 (1/4)

❑파일의 크기 제한

①

```c
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(int argc, char *argv[])
{
    int nFd = -1;
    int nFileSize = 1024;
    int nOffset = 0;
```

```
#<domain>      <type>  <item>         <value>
#
test             soft    fsize          2000
```

[파일의 최대크기 제한 설정 예]

②

```c
    if (argc != 3)
    {
        printf("Usage: %s <filename> " \
        "<filesize(KB)>.\n", argv[0]);
        return -1;
    }

    nFileSize = atoi(argv[2]) * 1024;

    nFd = open(argv[1], O_WRONLY | O_CREAT, 0666);
    for (nOffset = 0; nOffset < nFileSize; nOffset++)
    {
        char ch = 'A';
        if (write(nFd, &ch, 1) < 0)
            break;
    }
    close(nFd);

    printf("We wrote %d bytes.\n", nOffset);

    return 0;
}
```

[파일명: **FileSize.c**]

2. 제한 1 (2/4)

인덕대학교

❑ 파일의 접속 개수 제한 (1/2)

1

```
1  #include <sys/types.h>
2  #include <sys/stat.h>
3  #include <fcntl.h>
4  #include <stdio.h>
5  #include <stdlib.h>
6  #include <string.h>
7  #include <unistd.h>
8
9  #define MAX_FDS 1024
```

```
#<domain>      <type>  <item>        <value>
#
test            soft    nofile         20

#*              soft    core           0
```

[파일의 접근 개수를 제한 예]

```
11 int main(int argc, char *argv[])
12 {
13 2   int p_nFd[MAX_FDS];
14     int nFiles = 10;
15     int nCount = 0;
16     char p_Buffer[BUFSIZ];
17
18     if (argc != 3)
19     {
20         printf("Usage: %s <filename> <File Count> ", argv[0]);
21         return -1;
22     }
23
24     nFiles = atoi(argv[2]);
25     if(nFiles>1024)
26         nFiles = 1024;
27
28     for (int i = 0; i < nFiles; i++)
29     {
30         p_nFd[i] = -1;
31     }
```

[파일명: **FileOpenCount.c (1/2)**]

2. 제한 1 (3/4)

인덕대학교

❑ 파일의 접속 개수 제한 (2/2)

```
#<domain>      <type>  <item>         <value>
#
test             soft    nofile          20

#*               soft    core             0
```

[파일의 접근 개수를 제한 예]

```
28  for (int i = 0; i < nFiles; i++)
29  {
30      p_nFd[i] = -1;
31  }
32
33  for (nCount = 0; nCount < nFiles; nCount++)
34  {
35      memset(p_Buffer, 0, BUFSIZ);
36      sprintf(p_Buffer, "%s_%d.txt", argv[1], nCount);
37      p_nFd[nCount] = open(p_Buffer, O_WRONLY | O_CREAT, 0666);
38      if (p_nFd[nCount]<0)
39          break;
40  }
41
42  printf("We Opened %d.\n", nCount);
43
44  for (int i = 0; i < nCount;i++)
45      close(p_nFd[i]);
46
47  return 0;
48  }
49
```

3

[파일명: **FileOpenCount.c (2/2)**]

2. 제한 1 (4/4)

❑ 메모리 크기 제한 (2/2)

```
#<domain>      <type>  <item>         <value>
#
test            soft    data            10240
test            hard    data            10240
```

[메모리 크기 제한 예]

1

```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(int argc, char *argv[])
{
    char *p;
    int nDataSize = 1024;
    int nOffset = 0;
```

2

```c
    if (argc != 2)
    {
        printf("Usage: %s <datasize(KB)>.\n",
        argv[0]);
        return -1;
    }

    nDataSize = atoi(argv[1]);

    for (nOffset = 0; nOffset < nDataSize; nOffset++)
    {
        p = (char *)malloc(1024);
        if(p==NULL)
            break;
    }

    printf("We allocated %d kilo-bytes.\n", nOffset);

    getchar();

    return 0;
}
```

[파일명: DataSize.c]

수고하셨습니다.