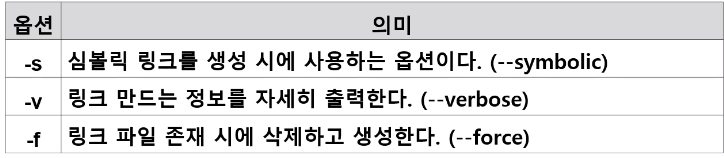
Report

서버관리실무

**제목 : 10주차 과제**

**학과 : 컴퓨터소프트웨어과  
학번 : 2017100301  
이름 : 이상남  
 담당교수 : 김형철 교수님  
제출일 : 2020.11.03**

1. **파일링크  
   리눅스에서 파일이나 디렉터리를 생성하면 i-node라는 번호가 임의로 부여되고, 이 번호를 기준으로 관리된다.  
   이번호는 is-I 명령으로 확인할 수 있는데, 파일명이 다르더라도 이 번호가 같다면 내부적으로는 같은 파일로 인식된다.  
   하나의 파일을 여러 개의 이름으로 관리하거나 디렉터리의 접근경로를 단축하는 형태를 링크라고 부르고 ln명령을 이용해서 만들수 있다.**
2. **하드링크  
   하드링크는 하나의 동일한 파일을 디스크의 다른 곳에 배치( 복사와 같은 형태) 하여 여러 이름으로 사용되는 형식이다.  
   하드링크는 파일에만 부여할 수 있는데, 리눅스 초기에는 sticky\_bit가 설정되지 않는 공유 디렉터리에서 사용하였다.  
   특징**
3. **ls -I 명령으로 i-node 번호를 확인하면 원본과 링크 파일의 번호가 같다.**
4. **원본과 링크 파일의 내용과 파일의 크기가 같다.**
5. **ls -l 명령시에 출력되는 링크의 숫자가 올라간다.**
6. **원본이나 링크 파일 중에 어떠한파일을 수정해도 같이 반영된다.**
7. **원본 파일을 삭제해도 링크 파일은 아무런 영향을 받지 않는다.**
8. **하드 링크는 파일만 설정가능하고, 동일한 파일 시스템에서만 사용가능 하다.**
9. **파티션이나 디스크 드라이브를 가로질러 사용할 수는 없다.**
10. **심볼릭 링크  
    하나의 파일을 여러 이름으로 가리키게 하는 것으로 원본과 생성된 링크 파일은 완전히 다른 파일로 관리된다.  
    파일이나 디렉터리에 모두 사용 가능하나 입ㄹ반적으로 디렉터리의 경로 단축이나 변경에 사용된다.  
    특징**
11. **ls -I 며열ㅇ으로 i-node 번호를 확인하면 원본과 생성된 링크 파일의 번호가 다르다.**
12. **생성된 링크 파일의 크기가 매우 작다.**
13. **ls -l 명령 시에 출력되는 권한 영역의 맨 앞쪽에 ‘ㅣ’ 이라고 표시된다.**
14. **원본이나 링크 파일 중에어떠한 파일을 수정해도 같이 반영된다.**
15. **원본 파일을 삭제하면 링크 파일은 아무런 구실을 하지 못한다.**
16. **딜게터리에 링크 파일을 생성하면 윈도의 바로가기나 단축아이콘의 기능과 같다.**
17. **생성되는 링크 파일의 퍼미션값이 777로 표시되나, 이 값은원본 파일의 퍼미션과는 무관하다.**
18. **ln : 하드링크 또는 심볼릭 링크를 생성하는 명령이다. 옵션없이 사용하면 하드링크가 생성되고,   
    -s옵션을 사용하면 심볼릭 링크를 생성한다.  
    [사용법] $ln [option] 원본 대상파일명   
    [주요 옵션]**



**[사용 예]  
 $ln joon.txt j -> joon.txt라는 파일의 하드링크 파일인 j를 현재 디렉터리에 생성한다.  
 $ln -s joon.txt j -> joon.txtx 라는 파일의 심볼릭 링크 파일인 j 를 현재 디렉터리에 생성한다.  
 $ln -s/etc/xinetd.d x ->/etc/xinetd.d의 심볼릭 링크 파일인 x 를 현재 디렉터리에 생성한다.**

1. **프로세스의 정의**

* **리눅스는 한 번에 수 백여 개 이상의 프로그램을 저장하고 동시에 수행할 수 있다.**
* **프로그램이란 어떤 문제를 해결하기 위해 사용되는 명령어나 유틸리티의 집합이라고 할 수 있따.**
* **보통 프로그램을 설치하면 하드디스크에 저장되고, 특정한 목적을 위해 프로그램을 실행하면 메모리(보통RAM이라 ㅂ부르는 주기억장치)에 상주하게 된다.**
* **실행중인 프로그램을 프로세스라 하고, 실행 시에 번호(PID)가 할당되어 관리된다.**
* **사용자의 입력에 관계없이 실행되는 백그라운드 프로세스와 명령 입력 후 수행종료 까지 기다려야 하는 포어그라운드 프로세스가 있다.**

1. **프로세스에 대한 다양한정의**
2. **실행 중인 프로그램**
3. **PCB를 지닌 프로그램**
4. **프로그램 카운터를 지닌 프로그램**
5. **능동적 개체로 순차적으로 수행하는 프로그램**
6. **프로세스의 생성**

* **하나의 프로세스가 다른 프로세스를 실행하기 위한 시스템 호출방법에는 fork와 exec가 있다.**
* **fork는 새로운 프로세스를 위해 메모리를 할당받아 복사본 형태의 프로세스를 실행하는 형태**
* **exec는 원래의 프로세스를 새로운 프로세스로 대체하는형태로 호출한 프로세스의 메모리에 새로운 프로세스의 코드를 덮어씌워 버린다.**

1. **프로세스의 종류**

* **보통 셸에서 명령을 실행하면 해당프로세스가 종료될 때까지 기다려야 하는데, 이러한 프로세스를 포어그라운드 프로세스라 한다.**
* **백그라운드 프로세스는 말 그대로 눈에 보이지 않게 뒤에서 실행시키는 방식이다.**
* **실행시키는 방법은 기존의 포어그라운드 프로세스실행 명령 뒤에 & 만 추가로 붙여서 실행하면 되고, 다중 작업을 수행할 때 유용하다.**