|  |
| --- |
| * Ch\_1 \_ 셸의 기본과 작성법 * Ch\_2 \_ 변수, 파라미터, 조건문 * Ch\_3 \_ 반복문, 함수, set등 * Ch\_4 \_ 텔넷 서버 * Ch\_5 \_ OpenSSH 서버 * Ch\_6 \_ XRDP 서버 * 셸의 기본과 작성법 * 우분투의 bash 셸 (터미널) * 기본 셸은bash(Bourne Again Shell:’배시 셸’) * bash 셸의 특징 * Alias 기능(명령어 단축 기능) * History 기능(위/아래 화살표키) * 연산 기능 * job Control 기능 * 자동 이름 완성 기능 (탭키) * 프롬프트 제어 기능 * 명령 편집 기능 * 셸의 명령문 처리 방법 * (프롬프트) 명령어 [옵션…][인자…] * 예) # rm -fg /mydir * 환경 변수 * “echo $환경변수이름” 으로 확인 가능 * “export 환경변수=값” 으로 환경 변수의 값을 변경 * 주요 환경변수        * HOME 과 PATH 사용 확인 * 셸 스크립트 프로그래밍 * c언어와 유사하게 프로그래밍이 가능 * 변수, 반복문, 제어문 등의 사용이 가능 * 별도로 컴파일하지 않고 텍스트 파일 형태로 바로 실행 * vi나 gedit으로 작성이 가능 * 리눅스의 많은 부분이 셸 스크립트로 작성되어 있음      * 셸 스크립트의 작성과 실행      * 실행방법  1. “sh <스크립트파일>”로 실행 2. “chmod +x <스크립트 파일>” 명령으로 실행 가능 속성으로 변경한 후에, “./<스크립트파일>” 명령으로 실행        * gedit로 name.sh을 생성 뒤 작성 후 실행 * 사용자이름과 홈 디렉터리가 나타나는 기능 코딩 * sh <스크립트파일> 로 실행 (sh name.sh)      * 파일 속성 변경 후 확인 * chmod +x [파일이름] 명령어로 파일 실행 속성 추가 * 변수의 기본 * 변수를 사용하기 전에 미리 선언하지 않으며, 변수에 처음 값이 할당되면서 자동으로 변수가 생성 * 모든 변수는 ‘문자열(String)’로 취급 * 변수 이름은 대소문자를 구분 * 변수를 대입할 때 ‘=’ 좌우에는 공백이 없어야 함      * ‘=’ 오른쪽이 주입이 된다. * 출력할때 $ (대입할 때는 붙이지 않는다) * 대입할 문자에 공백이 있을 경우 “ ” 사용 * 문자열로만 취급하기 때문에 testval=7+5 가 계산되지 않고 그대로 출력 * 변수의 입력과 출력 * ‘$’ 문자가 들어간 글자를 출력하려면 ‘ ‘ 로 묶어주거나 앞에 ‘\’ 를 붙임 * “ “ 로 변수를 묶어줘도 된다.        * vi 에디터 사용 * 숫자 계산 * 변수에 대입된 값은 모두 문자열로 취급 * 변수에 들어 있는 값을 숫자로 해서 +,-,\*,/ 등의 연산을 하려면 expr을 사용 * 역따옴표를 사용 * 수식에 괄호 또는 곱하기(\*)는 그 아페 꼭 역슬래쉬(\) 붙임        * gedit 에디터 사용 * 파라미터 변수 * 파라미터(Parameter) 변수는 $0, $1, $2 … 의 형태를 가짐 * 전체 파라미터는 $\*로 표현   예)           * 기본 if 문 * 형식   if [조건]  then  참일 경우 실행  fi     * “[ 조건 ]”의 사이의 각 단어에는 모두 공백이 있어야 한다. * if~else 문 * 형식   if [ 조건 ]  then  참일 경우 실행  else  거짓인 경우 실행  fi     * 조건문에 들어가는 비교 연산자        * 파일과 관련된 조건        * case~esac 문 * if 문은 참과 거짓의 두 경우만 사용 (2중분기) * 여러 가지 경우의 수가 있다면 case문 (다중문기)        * AND, OR 관계 연산자 * and는 ‘-a’ 또는 ‘&&’를 사용 * or는 ‘-o’또는 ‘||’를 사용        * 반복문 - for~in문 * 형식   for 변수 in 값1 값2 값3 …  do  반복할 문장  done     * 현재 디렉터리에 있는 셸 스크립트 파일(\*.sh)의 파일명과 앞 3줄을 출력하는 프로그램      * 반복문 - while 문 * 조건식이 참인 동안에 계속 반복      * 1에서 10까지의 합계를 출력 (‘반복문 - for’ 내용과 동일)      * 비밀번호를 입력받고, 비밀번호가 맞을 때까지 계속 입력받는 스크립트        * until 문 * while문과 용도가 거의 같지만, until문은 조건식이 참일때까지(=거짓인 동안) 계속 반복한다 * while2.sh를 동일한 용도로 until문으로 변경하려면 4행을 다음과 같이 변경하면 된다. * until [ $i -gt 10 ] * break, continue, exit, return 문 * bradk는 주로 반복문을 종료할 때 사용되며, continue는 반복문의 조건식으로 돌아가게 함. exit는 해당 프로그램을 완전히 종료함. return은 함수 안에서 사용 될 수 있으며 함수를 호출한 곳으로 돌아가게 함 * 사용자 정의 함수      * 함수의 파라미터 사용 * 형식   함수이름 () { → 함수를 정의  $1, $2 … 등을 사용  }  함수이름 파라미터1 파라미터2 … → 함수를 호출     * eval * 문자열을 명령문으로 인식하고 실행      * export * 외부 변수로 선언해 준다. 즉, 선언한 변수를 다른 프로그램에서도 사용하 수 있도록 해줌      * printf * c언어의 printf() 함수와 비슷하게 형식을 지정해서 출력      * set과 $(명령어) * 리눅스 명령어를 결과로 사용하기 위해서는 $(명령어) 형식을 사용 * 결과를 파라미터로 사용하고자 할 때에는 set과 함께 사용      * shift * 파라미터 변수를 왼쪽으로 한 단계씩 아래로 쉬프트(이동) 시킨다      * 텔넷 서버 개요 * 오랫동안 전통적으로 사용되어 온 원격 접속 방법 * 보안에 취약 * 리눅스 서버에 텔넷 서버를 설치하고 나면, 원격지에서 접속할 pc에는 텔넷 클라이언트 프로그램이 필요 * 원격지의 pc(텔넷 클라이언트)에서 접속하게 되면 서버 앞에 앉아서 직접 텍스트 모드로 작업하는 것과 완전히 동일한 효과      * 텔넷 서버 구축 * 원격지에서 서버 접속할 경우 필요 * 텔넷 서버 설치 과정 요약      * 서버에 접속하기 위해서는 꼭 클라이언트 프로그램 필요 * 서버가 리눅스라고 클라이언트도 리눅스일 필요는 없음 * 각각의 서버 프로그램은 자신에 맞는 별도의 클라이언트 프로그램이 필요\      * 텔넷 서버 설치 (apt -y install xientd telnetd)      * 설정 파일 편집 (/etc/xinetd.d/teinet 파일 편집)      * 텔넷 전용 사용자 생성 * root사용자로 이용하면 위험하기 때문에 adduser 명령을 사용하여 생성 (teluser, passwd= teluser 생성)      * 텔넷 서비스 가동 // systemctl restart xinetd      * 방화벽 설정(포트 열기) // ufw allow 23/tcp (텔넷은 23번 포트 사용)      * 서버 컴퓨터로 접속 테스트 * 서비스 작동 상태를 확인하기 위해 서버 자기 자신에게 접속 테스트를 진행 함으로써 문제가 발생할 때 폭을 줄일 수 있다.      * 서버에 클라이언트가 접속하기 위해선 그 서버 전용의 클라이언트가 필요하다 (ex, 텔넷서버 < 텔넷클라이언트 웹서버 < 웹클라이언트 , db서버 < db 클라이언트 등)      * winclient에서 텔넷 클라이언트 적용 * 앱 및 기능 > 프로그램 기능 > windows 기능에서 텔넷 클라이언트 확인 후 재부팅      * windows powershell 에서 텔넷 서버로 ping 응답 확인 * ping [ip주소]      * 클라이언트에서 접속      * 접속 후 사용자 정상 확인 // username: teluser passwd: teluser      * 텔넷 클라이언트를 windows용을 사용하면 한글이 깨지는 상황        * putty 사용하여 한글 깨짐 없이 사용 가능 (필수x) * OpenSSH 서버 * 텔넷과 용도는 동일하지만, 보안이 강화 * 텔넷과 거의 동일하지만 데이터를 전송할 때 암호화를 한다는 점이 다름      * 원격지에서 보안이 강화된 서버 접속할 경우 필요 * Opeenssh 서버 설치 과정 요약        * ssh 서버 설치 // apt install openssh-server      * ssh 서버 재부팅(적용) // systemctl restart ssh      * ssh 서버 상태 확인 // systemctl status ssh      * 방화벽 설정 // ufw allow 22/tcp (openssh은 22번 포트)      * windows 클라이언트에서 접속 (본인은 putty사용했으나 터미널도 같음) * XRDP 서버 * X 윈도우 환경으로 원격접속을 사용하고 싶을 때 * Windows의 ‘원격 데스크톱 연결’ 프로그램을 사용해서 리눅스에 그래픽 환경으로 접속      * 원격지에서 X 윈도우 모드로 접속할 경우 필요 * XRDP 서버 설치 과정 요약        * XRDP 서버 설치 // apt install xrdp      * 서비스 시작 * xrdp 서버 재부팅 (systemctl restart xrdp) * xrdp 서버 상태 확인 (systemctl status xrdp)      * 항시 가동 적용 * 전원이 꺼지면 서버도 off되어 다시 위와 같은 과정을 거쳐야 하기 때문에 (보통 서버는 항시 가동) systemctl enable xrdp 명령어로 항시 가동을 적용      * 방화벽 설정 (ufw allow 3389/tcp) // xrdp는 3389포트      * Windows에서 원격 데스크톱 연결      * ip연결 후 로그인 (name: ubuntu passwd: ubuntu)      * 원격 연결 성공 * 3가지 원격 접속 서버의 비교 * 비교표      * 결론 * SSH를 기본적으로 사용하고, XRDP 서버는 설정만 해놓고 가동하지 않는다. 원격지에서 SSH로 서버를 관리하다가, X 윈도우 접속이 필요할 경우에는 접속된 SSH 접속 창에서 XRDP 서버를 구동하고 Windows의 원격 데스크톱으로 접속해서 사용 * 텔넷 서버는 보안이 철저한 회사 내의 네트워크에서만 사용하면 무난함 |