1주차 결과보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20211558 이름: 윤준서

**1. 목적**

수업시간에 실습한 쉘 프로그램을 확장해본다.

**2. 숙제풀이 결과**

2-1

|  |
| --- |
| **#!/bin/bash** #쉘 스크립트 시작 전 선언한다.  **if [ $# -lt 1 ];then** #입력한 값이 아무것도 없는 경우 출력되는 문장을 만든다  **echo "Usage : phone searchfor [...searchfor]"**  **echo "(You didn't tell me what you to earch for.)"**  **exit 1 #end the script** #이 줄을 처음에 쓰지 않았을 때 문장이 출력되고  프로그램이 종료되지 않는 현상이 일어났다. 이를 통해  문장이 출력된 후 쉘을 종료하도록 한다.  **fi**  **input="$1"** #입력한 값을 새로운 문자열 변수로 만든다. 첫 번째 인자만을 우선 변수로  받는다.  **for i in $\*** #쉘 스크립트 실행 시 입력한 값 전체를 반복문에 이용한다. i 한 번당  입력한 값의 한 인자이다.  **do**  **input+="|"**  **input+=$i** #우리가 입력한 값을 공백을 없애고 | 를 붙여서 하나의 긴 문자열이  이어진 형태로 변수를 만든다. 후에 egrep 을 통한 데이터 값을 찾을  때 data 파일에 있는 데이터들이 | 가 붙여지면서 쓰여져 있기에  수월한 데이터 찾기가 가능해진다.  **done**  **egrep -i $input data | awk -f display.awk**  #"egrep -i [변수명] [파일명]" 을 이용해 입력한 데이터와 같은 값을  data 파일에서 찾는다. 이후 "| awk -f [awk 파일명]" 을 통해 찾은 값을  awk 파일을 참조, 실행한다. 실행할 때 display.awk 파일을 사용한다.  그럼 display.awk 내의 형식에 알맞은 형태로 데이터들이 출력된다. |

2-2

and 형식과 or 형식을 사용하기 위해 사용되는 입력의 형식으로 다음과 같다.

and : && or : ||

and 형식과 or 형식을 프로그램에 구현하기 위해선 if 문을 이용한 조건문이 필요하다.

and 형식의 경우 두 개 이상의 조건을 모두 만족시켜야 하므로 조건문 속에 조건문을 사용한다.

순서는 만들고자 하는 프로그램에 따라 나뉜다.

앞선 if 문을 만족시키면 그 안의 실행에 if 문이 있을 것이고 이 또한 만족해서 두 번째 if 문 안의 조건을 실행한다면 두 개의 조건을 만족시키는 and 형식의 조건문이 완성된다.

또는 한 조건문에 && 을 이용하여 두 개의 조건을 동시에 판별하는 if 문을 만들 수 있다.

or 형식의 경우 동시에 만족이 아닌, "또는" 에 해당하는 형식이기에 조건문을 다단계에 걸쳐서 입력하는 것이 아닌, 조건문의 또 다른 형식 else if (쉘 스크립트에선 elif 로 쓰인다) 와 else 를 이용한다. if 조건문을 만족시키지 못했을 경우 넘어가는 새로운 조건문이다.

이 또한 순서는 만들고자 하는 프로그램에 따라 달라진다. 순서에 따라 본인이 의도한 실행 결과가 달라질 수 있기에 주의해야 한다.

else if(elif) 와 else 의 차이점으로는 영어로 설명하자면 another 과 the other 의 차이와 같다. else if(elif) 의 경우 만족시키지 못했을 경우에 넘어가는 새로운 조건문들이 2개 이상 있을 때 사용한다. 그리고 else 의 경우 마지막 순서의 새로운 조건문에 사용한다.

다음은 가시적인 이해를 위한 간단한 쉘 스크립트이다.

어떤 숫자를 입력했을 때, 그것이 "홀수이며 3의 배수다" 와 "짝수이다" 를 판별한다.

|  |
| --- |
| **#!/bin/bash**  **read number**  **if [ (($number) % 2) -eq 1 ] && [ (($number) % 3) -eq 0 ];** #&& 를 이용해 한번에  **then ehco "number 은 홀수이며 3의 배수다"** and 의 조건을 판별한다.  #위의 조건문을 만족하지 않는 경우 or 의 정의에 맞게 다음 조건문을 판별한다.  **elif [ (($bumber) % 2) -eq 0 ];**  **then echo "number 은 짝수이다"** |