2020년도 2학기 컴퓨터공학설계및실험I

1주차 결과 보고서

20170175 김태안

1. 실습 목적

문자열을 검색할 수 있는 쉘 프로그램을 작성한다. 수업시간에 실습한 쉘 프로그램을 확장해본다.

1. 실습 구현 내용

**phone**

입력 받은 인자를 (인자)|(인자)|…형태로 다듬은 뒤, mydata 파일에서 인자를 검색한다. 검색은 OR 연산자를 기준으로 이뤄진다. 검색 결과는 display.awk 파일을 통해 출력한다. 인자가 없을 경우 예외처리를 한다.

|  |
| --- |
| if [ $# -eq 0 ]  then  echo "Usage: phone searchfor [...searchfor]"  echo "(You didn't tell me what you want to search for.)"  else  for arg in $@  do  args="$args|$arg"  done  args=${args: 1}  egrep $args mydata | awk -f display.awk  fi |

**display.awk**

입력 받은 인자를 형식에 맞게 출력한다.

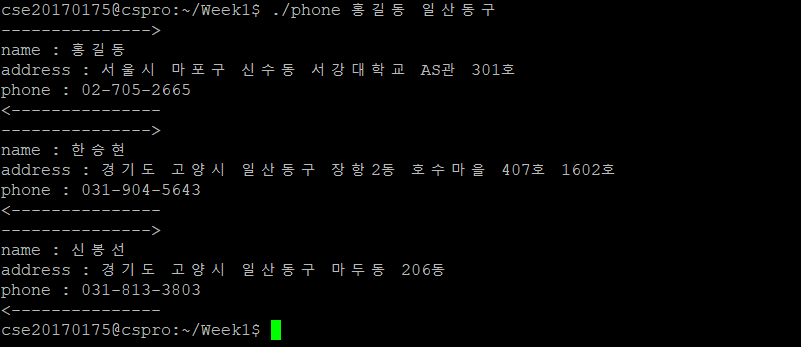
|  |
| --- |
| BEGIN{FS="|"}  {print "--------------->";}  {print "name : "$1;}  {print "address : "$2;}  {print "phone : "$3;}  {print "<---------------";} |

1. 실습 환경

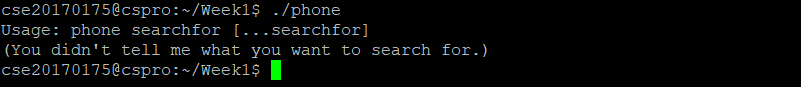
cspro.sogang.ac.kr

OS: Ubuntu 16.04.2 LTS (GNU/Linux 4/4/0-184-generic x86\_64)

1. 실습 결과 및 분석



phone을 통해 홍길동, 일산동구를 검색하면 홍길동과 일산동구를 포함한 데이터를 모두 출력한다.



인자를 입력하지 않으면 함수의 역할과 에러 메시지를 출력한다.

**OR 형식이 아닌 AND 형식으로 검색을 하기 위한 프로그램**

위 프로그램을 OR가 아닌 AND 형식으로 검색하는 프로그램으로 변경해보자. AND를 검색하기 위해 텍스트를 (인자)|(인자)… 형식이 아닌 (인자).\*(인자)… 형식으로 변경한다. .\*는 정규 표현식이다.

|  |
| --- |
| if [ $# -eq 0 ]  then  echo "Usage: phone searchfor [...searchfor]"  echo "(You didn't tell me what you want to search for.)"  else  for arg in $@  do  args="$args.\*$arg"  done  args=${args: 2}  egrep $args mydata | awk -f display.awk  fi |



위 프로그램을 실행하면 “신봉선 경기도 고양시”를 검색하면 모두 포함한 결과 1개를, “경기도 고양시”를 검색하면 경기도 고양시가 포함된 결과 2개를 출력한다.

**AND 형식과 OR 형식을 동시에 지원하는 프로그램 설계**

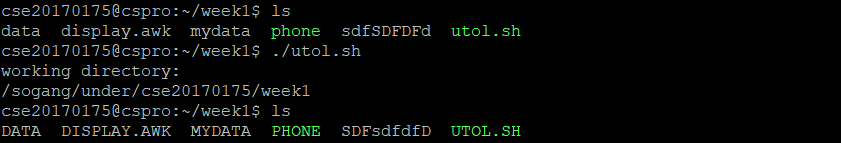
phone의 첫 인자로 검색 내용이 아닌 AND/OR를 입력 받는다. 이후 나머지 인자를 모두 검색 내용으로 인식하고 AND/OR 여부에 따라 다른 연산을 한다.

1. 과제

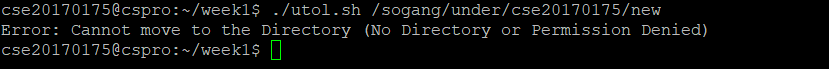
**디렉토리 내부의 모든 파일의 대소문자를 반전하는 프로그램**

인자로 경로를 받아 이동한 뒤, 디렉토리의 모든 파일을 읽어 그 대소문자를 변환한다. 변환은 정규표현식을 이용한다.

|  |
| --- |
| if [ $# -eq 0 ]  then  path=`pwd`  else  path=$@  if [ -d $path -a -w $path ]  then  cd $path  else  echo "Error: Cannot move to the Directory (No Directory or Permission Denied)"  exit 0  fi  fi  echo "working directory:"  echo $path  for i in \*  do  filename=`echo $i | tr '[a-zA-Z]' '[A-Za-z]'`  mv $i $filename  done |



프로그램을 실행하면 모든 파일의 대소문자가 반전된다. 이때, utol.sh 자체도 반전되어 다시 사용하려면 UTOL.SH를 실행해야 한다.



경로가 없거나 권한이 없는 경로는 에러 메시지를 출력한다.

1. 결론

이번 실습에서는 UNIX 환경의 기본 사용법과 vim, 그리고 쉘 프로그래밍에 관해 실습했다. UNIX 환경은 윈도우와는 다른 명령어를 통해 동작한다. 또한, GUI가 없는 경우 명령어만을 이용해 운영체제를 사용할 수 있다. 텍스트 편집기 중 하나인 vim은 이러한 환경에서 사용할 수 있도록 명령어를 이용해 텍스트를 편집하고 저장할 수 있다. 쉘은 그 자체로 프로그래밍 언어인 명령어 처리기이다. 따라서 다양한 함수를 이용해 명령어를 처리하며, 명령어의 집합을 파일로 만들어 실행할 수도 있다.