**Data Structure Term Project**

**Final Report**

**2016320205 이지혜**

**2015410072 김재현**

**~~2014250413 이상화~~**

**· 서론**

**· 주제 선정 이유**

**현대 사회에서 SNS의 이용량이 급격히 증가함에 따라(그림1), 유저들의 동향과 관심사를 분석하기 위해서 SNS를 분석하는 것이 필요불가결하게 되었다. 이에, 31조는 SNS상의 친구관계와 게시글을 분석하는 프로그램을 제작하게 되었다.**

**(통계자료 삽입)**

**· 개발 내용에 대한 간단한 소개**

**모바일 디바이스 환경에서 SNS의 게시글의 키워드와 유저의 영향력을 분석할 수 있다.**

**(그림 삽입)**

**· 기존 시스템 조사**

**X, Y, Z와 같은 기존의 SNS분석 시스템은 단순히 하나의 SNS만 분석할 수 있으며, 다수의 SNS를 분석할 수 없다. 또한 플랫폼이 대단히 한정적이다.**

**· 기존 시스템과의 차별성**

**하나의 SNS만 분석할 수 있는 기존 시스템과 달리 여러 SNS를 분석할 수 있다. 또한 기존에 있던 여러 SNS분석 시스템은 PC환경에 국한되었으나, 본 어플리케이션은 모바일 디바이스 환경을 제공한다.**

**· 결론**

**· 역할분담**

**이상해씨 : **

**이지혜 : 갓갓갓(버스 부릉부릉)**

**김재현 : 이거 쓰고 있음 + C언어 짬**

**· 아쉬운 점 및 향후 발전 방향**

**현재 프로그램은 단순히 단어를 분석하는데 그치기 때문에 심도있는 기호파악이 불가능하다. 예를 들어, 다른 유저들보다 영향력이 큰 유저가 끼치는 차별적인 영향력에 대한 다각적인 분석이나, 가장 많이 언급된 단어들 간의 상관관계를 파악하는 것과 같은 고차원적인 분석이 불가능하다. 따라서 키워드 간의 상호작용과 상관관계를 파악하여 더 심도있는 맥락 파악과, 유저의 영향력에 따른 가중치를 부여하여 단어별 영향력 기여도 등을 정확히 파악하는 방향으로 나아가야한다. 이 과정을 통해 사용자의 의사결정에 보다 구체적이고 유의미한 영향을 끼칠 수 있다.**

**user.h**

유저에 관련된 구조체와 함수를 정의한 헤더

|  |  |  |  |
| --- | --- | --- | --- |
| #pragma | once |  | 중복 컴파일 방지 |
| #define | MAX\_USER | 7500 | 저장할 수 있는 최대 유저수 |
|  |  |  |  |
| typedef | struct | User | 유저의 정보를 가지고 있는 구조체 |
|  | int | id\_num | 해당 유저만이 가지고 있는 고유번호 |
|  | char | sign\_up\_date[40] | 가입일을 나타내는 문자열 |
|  | char | name[20] | 유저의 닉네임 |
|  | Word | \*tweets | 유저가 트윗한 내용을 가리키는 포인터 |
|  | User | \*friends | 유저가 등록한 친구를 가리키는 포인터 |
|  |  |  |  |
| typedef | struct | UserType | User 구조체의 정보를 가지고 있는 구조체 |
|  | int | count | User의 수 |
|  | User | \*user[MAX\_USER] | User를 가리키는 포인터 배열 |
|  |  |  |  |
| typedef | struct | Word | 트윗과 관련된 정보를 가지고 있는 구조체 |
|  | char | upload\_date[40] | 트윗한 날짜를 나타내는 문자열 |
|  | char | tweet[270] | 트윗한 내용을 나타내는 문자열 |
|  | Word | \*next\_tweet | 해당 유저가 한 다른 트윗을 가리키는 포인터 |
|  |  |  |  |
| void | init(UserType \*u) | | UserType 구조체에 정보를 입력하는 함수 |
| void | init\_userType(UserType \*u) | | UserType 구조체를 초기화하는 함수 |
| void | insert\_user(UserType \*u) | | UserType 구조체에 User를 삽입하는 함수 |
| void | add\_friends(UserType \*u) | | UserType 구조체에 User에 유저가 등록한 친구를 입력하는 함수 |
| void | add\_tweets(UserType \*u) | | UserType 구조체에 User에 유저가 한 트윗을 입력하는 함수 |
| int | compare (const void \*a, const void \*b) | | qsort를 위한 비교함수 |

**user.c**

user.h를 구현하는 C파일

|  |  |  |  |
| --- | --- | --- | --- |
| #include | <stdio.h> | | 표준 입출력을 위한 헤더 |
| #include | <stdlib.h> | | 메모리를 할당하기 위한 헤더 |
| #include | <string.h> | | 문자열 복사와 메모리 복사를 위한 헤더 |
| #include | "user.h" | | 구현할 헤더 |
|  |  |  |  |
| void | init(UserType \*u) | | UserType 구조체를 초기화하고 유저, 친구, 트윗 정보를 입력하는 함수 |
|  |  | |  |
| void | init\_userType(UserType \*u) | | UserType의 User 수를 0으로 초기화 |
|  |  |  |  |
| void | insert\_user(UserType \*u) | | user.txt 파일에서 유저 정보를 불러와 User 구조체에 입력한 뒤, UserType의 User 포인터 배열에 대입하고 User 수를 증가시킨다. 모든 User 입력이 끝나면 User 포인터 배열을 qsort로 User 구조체의 id\_num에 따라 오름차순으로 정렬한다. |
|  |  |  |  |
| int | compare (const void \*a, const void \*b) | | qsort의 비교값을 위한 함수로, 인자의 a, b를 User\*\*로 형변환한 뒤, 각각의 id\_num을 비교하여 a의 id\_num이 크면 양수를, 같으면 0을, 작으면 음수를 반환한다. |
|  |  |  |  |
| int | find\_user\_index(UserType \*u, int id, int left, int right) | | UserType의 User 포인터 배열에서 인자인 id와 같은 id\_num을 가지는 유저의 이진탐색으로 찾은 뒤 해당 index를 반환한다. |
|  |  |  |  |
| void | add\_friends(UserType \*u) | | friend.txt 파일에서 친구 정보를 불러와 User 구조체에 입력한다. |
|  |  |  |  |
| void | add\_tweets(UserType \*u) | | word.txt 파일에서 트윗 정보를 불러와 User 구조체에 입력한다. |