**동계훈련 개별 프로젝트 보고서**

20133265 차동민

–Micro Scout-

1. **개요**

**마이크로 스카우트(Micro Scout)**은 ‘정찰자’ 라는 이름대로, 간략하게 리눅스 시스템 하에서의 정보를 확인하고 그에 대한 로그를 출력하거나 기록/저장한다. 간략한 정보들을 빠르게 파악하여 간략하게 기록하는 것이 역할이다. 기능들은 나중에 더 확장될 수도 있다.  
 프로그램의 인터페이스로는 타이틀 화면을 터미널에 출력한 후, 원하는 정보들을 선택하여 기록하게 하는 방식을 선택했다.

|  |
| --- |
|  |

예를 들면, 로그를 탐색하여 나름대로의 알고리즘으로 분석하여 임의의 파일에 그것을 기록할 수 있는 식으로 설계한다. 기록한 후 종료할 때 그 내용을 터미널에 출력해주는 기능을 넣을 수도 있다.

1. **구현할 기능의 예**

처음에 구상해본 몇 가지 기능들은 다음과 같다.

|  |
| --- |
| * 현재 가동중인 프로세스들 각각의 대략적인 중요도를 추측해내 기록하는 기능 * 메모리 공격(오버플로우)에 취약한 부분을 가지고 있는 파일을 찾아 기록하는 기능 * 시스템에 접속했던 사용자들의 동향을 기록하는 기능 * tmp 폴더 파일들 각각의 대략적인 중요도를 기록하는 기능 * 통신을 사용하는 프로그램을 찾는 기능 * 로그를 탐색해서 접속이 드문 사용자를 찾는 기능 * 그 외의 기능 |

꼭 위에 있는 기능이 아닌 다른 기능을 구현할 수도 있으나 대체적으로는 간략하게 찾아서 간략하게 기록하는 것을 이 프로젝트의 주 목적으로 정한다. 기능은 쓸모있을 수도 있고 쓸모없는 기능일 수도 있으나, 프로그램이 가진 기능들의 질적 가치보다는 프로그램의 구현 과정에 좀더 중심을 두고 설계하도록 한다.  
 최종적으로는 3가지 기능을 염두에 두고 있다. 로그 탐색, tmp 폴더 탐색, 프로세스 폴더 탐색 등의 기능을 구현하도록 한다.

1. **기능 구현의 어려움**

아직 학부 교과과정에서 운영체제나 시스템을 심도 깊게 다루지 않아 정보 탐색에 어려움을 겪었다. 그리고 C를 이용해 이들을 구현하는 데에 까다로움이 있어 쉘코드를 동원하는 것을 검토해 보았다. /proc 폴더와 /var/log의 로그 파일 등은 그냥 read 등을 이용해 읽기에는 환경이나 포맷이 까다로운 경우가 많지만, 터미널에서 명령어를 이용한다면 내용들을 쉽게 출력할 수 있다. 이 프로젝트는 C를 이용해 작성하기 때문에 C 내에서 sh 파일을 실행하는 방법을 이용한다.  
 프로세스들을 탐색하고자 한다면, 쉘코드를 이용해 /proc 폴더를 탐색하여 필요한 부분만 뽑아내 어딘가에 기록하는 명령어를 만든 후, 이 sh 코드 파일을 C 내부에서 실행한 후, 기록한 파일을 찾아서 읽어들이는 방식을 사용할 수도 있다.

|  |  |
| --- | --- |
| 이름 | 프로세스 중요도 탐색 |
| 기능 | * /proc 폴더를 탐색해 프로세스들의 사용 시간 등을 탐색해 중요도를 파악 * 혹은 프로세스의 실행자가 누구인지를 탐색해 중요도를 파악 * 프로세스의 기타 내용들을 이용해 중요도 파악 |
| 구현 | 미리 작성한 sh 파일을 사용한다. 이 sh 파일은 파악할 정보들을 read 함수가 읽기 쉽도록 어떤 파일에 기록하고, 그 파일을 C 코드 내에서 실행하여 데이터를 이용한다. |

|  |  |
| --- | --- |
| 이름 | tmp 폴더 중요도 탐색 |
| 기능 | * tmp 폴더 내부에 있는 파일들의 아이노드를 확인하여 정보를 탐색 * 접근 빈도수를 찾거나, 혹은 접근 권한이 다소 덜 중요한 파일을 탐색 * tmp 폴더는 임시로 생성된 로그 등의 파일이 위치하는데, 이들 중 필요없는 파일들이 어떤 것인지를 골라내는 알고리즘을 생각해 본다. |
| 구현 | 이는 리눅스 함수 중 아이노드에 접근하는 함수를 사용해 구현한다. 아이노드는 파일의 여러 정보를 가지고 있고, C에서 이 데이터에 접근해 정보들을 가져와 사용할 수 있다. |

|  |  |
| --- | --- |
| 이름 | 유저들의 기록 탐색 |
| 기능 | * 유저들의 접속 시간 등의 데이터를 모으고 정리한다. * 접속이 드문 이용자를 찾거나, 유저들 중 무언가 특별한 점이 있는지 찾아보는 기능 또한 검토해본다. * /var/log/에 들어있는 파일들을 참조한다. |
| 구현 | 미리 작성한 sh 파일을 이용한다. |

또한 프로젝트 기간이 짧다는 점 역시 설계 과정에서 어느 정도 한계에 부딪힐 가능성을 만들며, 이 보고서에 있는 내용을 전부 구현하지 못하고, 결과물의 내용을 축소시켜 완성하게 될 상황에도 처할 수 있으리라 예상하고 있다.

1. **과정**

어떤 식으로 중요도를 파악할지, 그에 대한 알고리즘이나 근거 조건들이 필요하다. 프로세스나 파일들의 정보들을 최대한 이용하여 그 중요도를 파악한다. 다만 이 프로젝트는 짧은 기간 동안에 구현해야 하므로 기능들이 미흡할 수 있으며, 이 중요도는 신뢰도가 높지 않을 수도 있다.  
 전체적인 인터페이스 기틀을 마련한 후, 각각의 기능들을 완성한다. 기능들이 공통적으로 사용하는 메소드나 자료 구조가 있다면, 헤더 파일에 넣는 것이 바람직할 것이고, 각각의 기능들을 별개의 C 파일로 만드는 것이 좋을 것이다.  
 현재로서 가장 중요한 과제는 다음과 같다.

|  |
| --- |
| * 필요한 정보들을 수집해 다루는 것 |
| * 수집한 정보들을 간략히 분석할 수 있는 알고리즘을 구상하는 것 |