**Pintos Project 1: User Program (1)**

담당 교수 : 김영재

조 / 조원 :

개발 기간 :

1. **개발 목표**

* **해당 프로젝트에서 구현할 내용을 간략히 서술.**

Pintos에서의 명령어 처리를 위해 Us

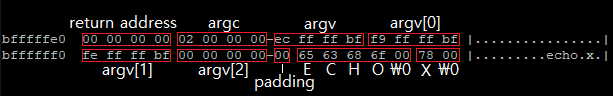
1. **개발 범위 및 내용**
   1. **개발 범위**

* **아래 항목을 구현했을 때의 결과를 간략히 서술**

1. Argument Passing
2. User Memory Access
3. System Calls
   1. **개발 내용**

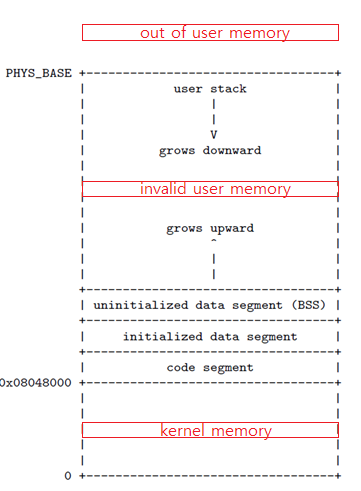
* **아래 항목의 내용만 서술 (기타 내용은 서술하지 않아도 됨.)**
* Argument Passing
  + 커널 내 스택에 argument를 쌓는 과정 설명

process.c의 load() 함수에서 setup\_stack() 함수에 의해 virtual memory가 생성이 된다. setup\_stack() 함수의 호출 이후 parsing된 argument를 user stack의 시작 지점인 PHYS\_BASE부터 쌓는다. 이때 4 byte씩 word-align하며 x86 calling convention에 맞게 user stack에 차례대로 쌓는다. PHYS\_BASE에서 4 떨어진 주소부터 argv[][ . . . ], padding bit, argv[], argv, argc, return address 순서대로 쌓는다. 이를 위해 목적지 주소값을 원하는 크기만큼 0으로 초기화하는 memset() 함수와 목적지 주소값을 소스 주소값에서 원하는 크기만큼 복사하는 memcpy() 함수를 사용한다. user stack에 argument를 모두 쌓고 해당 부분을 hex\_dump()로 출력하면 다음과 같다.



* User Memory Access
  + Pintos 상에서의 invalid memory access 개념을 간략히 설명

user side에서의 memory access는 user user side와 kernel side 중 user side로만 엄격히 제한된다. 따라서 user process는 kernel memory에 접근할 수 없다. 이와 달리 user memory 영역 안에 있지만 NULL 값을 갖거나 user memory 영역 밖으로 넘어가는 경우 또한 제한된다. 즉, kernel memory 영역으로의 접근, user memory 영역 안의 잘못된 접근, 그리고 user memory 영역 밖으로의 접근 세 가지 경우는 invalid memory access이다.



* + Invalid memory access를 어떻게 막을 것인지 설명

user process가 실행되는 동안 어떠한 memory으로의 접근이 이루어질 때는 항상 해당 memory가 유효한 접근인지 검사해야 한다. 그리고 pintos는 이를 위한 여러 함수들을 제공하고 있다. 먼저 kernel memory인지 판별하기 위해 is\_kernel\_vaddr() 함수를 이용한다. 반대로 user memory인지 판별하기 위해 is\_user\_vaddr() 함수를 이요한다. 위의 함수들은 인자로 들어온 포인터가 user memory인지 kernel memory인지 판단한다. 다음으로 user memory가 유효한지 판별하기 위해 pagedir\_get\_page() 함수를 이용한다. 위의 함수는 인자로 들어온 포인터가 실제로 user memory에 mapping 되어 있는지 판단한다. 위 함수들을 이용해 invalid memory access를 판단했다면 해당 process를 종료해야 한다.

* System Calls
  + 시스템 콜의 필요성에 대한 간략한 설명
  + 이번 프로젝트에서 개발할 시스템 콜에 대한 간략한 설명 (하나의 시스템 콜 당 최대 3문장으로 간략히 설명; 3문장을 넘길 정도로 길게 작성하지 말 것)
  + 유저 레벨에서 시스템 콜 API를 호출한 이후 커널을 거쳐 다시 유저 레벨로 돌아올 때까지 각 요소를 설명

1. **추진 일정 및 개발 방법**
   1. **추진 일정**

* **II. A.의 개발 범위를 포함하여 구현 내용에 대한 일정 작성**
  1. **개발 방법**
* **II. B.의 개발 내용을 구현하기 위해 어느 소스코드에 어떤 요소를 추가 또는 수정할 것인지 설명. (함수, 구조체 등의 구현이나 수정을 서술)**

1. **연구 결과**
   1. **Flow Chart**

* **II. B. 개발 내용에 대한 Flow Chart를 작성**

1. Argument Passing
2. User Memory Access
3. System Calls
   1. **제작 내용**

* **II. B. 개발 내용의 실질적인 구현에 대해 코드 관점에서 작성.**
* **구현에 있어 Pintos에 내장된 라이브러리나 자체 제작한 함수를 사용한 경우 이에 대해서도 설명.**
* **개발상 발생한 문제나 이슈가 있으면 이를 간략히 설명하고 해결책에 대해 설명.**

1. Argument Passing
2. User Memory Access
3. System Calls

* **이번 프로젝트에서 개발한 시스템 콜을 구현 관점에서 상세히 서술.**

1. Additional System calls

* **새로운 시스템 콜(fibonacci, max\_of\_four\_int)을 구현하기 위해 수정하거나 작성한 코드에 대해 서술**
  1. **시험 및 평가 내용**
* **fibonacci 및 max\_of\_four\_int 시스템 콜 수행 결과를 캡처하여 첨부.**