
Project 01

Implementing simple system call

Due date
2024. 03. 31. 11:59 pm

Project 01

- Project 1에서 해야 할 일은 아래와 같습니다.
 - getgid() (get grand parent process id) 시스템 콜 구현
 - User program 구현
 - Wiki 작성

getgpid()

- 부모 프로세스의 부모프로세스(조부모)의 pid를 반환하는 시스템 콜인 **getgpid()**를 구현합니다.
- getgpid() 시스템 콜은 유저 프로그램에서 사용될 수 있어야 합니다.

User Program

- 구현할 유저 프로그램의 이름은 "**project01**"입니다.
- 유저 프로그램은 과제를 수행한 사람의 학번, 프로세스의 id, 조부모 프로세스의 id를 출력합니다.
- 유저 프로그램 **project01**을 실행한 결과의 예시는 아래와 같습니다.

```
Booting from Hard Disk..xv6...
cpu0: starting 0
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2 inodestart 32 bmap start 58
init: starting sh
$ project01
My student id is 2022123123
My pid is 3
My gpid is 1
$
```

- 실행 결과는 제공된 예시와 다를 수 있습니다.

Evaluation

- **Completeness** : 명세의 요구조건에 맞게 xv6가 올바르게 동작해야 합니다.
- **Wiki & Comment** : 위키를 기준으로 채점이 진행되므로, 위키를 최대한 자세히 작성해야 합니다.
 - 이번 과제는 위키가 핵심인 과제입니다.
 - xv6에 대해 아무것도 모르는 사람이 위키만 보고 모든 것을 구현할 수 있을 정도로 모든 과정을 자세하게 적어주세요.
- **Deadline** : 마감 기한을 반드시 지켜야 하며, 마감 기한 이전에 마지막으로 제출된 코드를 기준으로 채점됩니다.
- **DO NOT SHARE AND COPY !!**

Wiki

- Wiki는 통상적으로 아래와 같은 요소를 포함합니다.
 - **Design**
 - 명세에서 요구하는 조건에 대한 구현 계획을 서술합니다.
 - **Implement**
 - 새롭게 구현하거나 수정한 부분이 기존과 어떻게 다른지, 해당 코드의 목적이 무엇인지에 대해 구체적으로 서술합니다.
 - **Result**
 - 컴파일 및 실행 과정과, 해당 명세에서 요구한 부분이 정상적으로 동작하는 실행 결과를 첨부하고, 동작 과정에 대해 설명합니다.
 - **Trouble shooting**
 - 과제 수행 과정에서 겪은 문제가 있다면, 해당 문제와 해결 과정을 서술합니다. 만약 문제를 해결하지 못했다면 어떤 문제였고 어떻게 해결하려 했는지 서술합니다.
- 더 필요한 내용이 있다면 작성해도 됩니다 ☺

Submission

- 구현한 코드와 wiki를 LMS 과제란에 제출합니다.
- Wiki 파일과 xv6-public 파일 전체를 한 디렉토리에 포함시킨 후, 압축하여 제출합니다.
- 제출할 디렉토리의 이름은 "OS_project01_[수업번호]_[학번]" 입니다.
- Wiki 파일의 이름은 "OS_project01_[수업번호]_[학번].pdf" 입니다.
 - 제출할 디렉토리의 양식 예시는 다음과 같습니다.
 - ex) OS_project01_12345_2022123123
 - xv6-public
 - OS_project01_12345_2022123123.pdf
 - 해당 디렉토리를 압축하여 OS_project01_12345_2022123123.zip 파일을 제출합니다.
- **제출 기한: 2024년 3월 31일 11:59 pm (No additional submission)**

Thank you

