

---

# Project 04 - Test Program

---

Implementing Copy-on-Write on xv6

**Due date**  
**2024. 06. 09. 11:59 pm**

# 기본 구조

---

- 다음의 코드들은 xv6 디렉토리와 Makefile에 추가하여 컴파일하고 실행합니다.
- 테스트시 실행할 프로그램 소스
  - test0.c
  - test1.c
  - test2.c
  - test3.c
- 출력 값이 ppt의 예시와 다르다면 그 이유를 위키에 적어주시길 바랍니다.
- 제공된 테스트 프로그램 이외의 테스트 프로그램이 존재할 수 있습니다.

# test0.c

---

- 4가지 시스템 콜인 countfp, countvp, countpp, countptp을 사용하여 현재 프로세스의 페이지 정보를 확인합니다.
- sbrk() 시스템 콜을 통해 추가적인 페이지가 알맞게 할당되었는지 확인합니다.

```
$ test0  
[Test 0] default  
ptp: 66 66  
[Test 0] pass
```

# test1.c

---

- fork() 시스템 콜을 통해 자식 프로세스를 생성합니다.
- 자식 프로세스와 부모 프로세스가 같은 물리 페이지를 가리키고 있는지 페이지 개수의 차이를 통해 확인합니다.

```
$ test1  
[Test 1] initial sharing  
[Test 1] pass
```

# test2.c

---

- fork() 시스템 콜을 통해 자식 프로세스를 생성합니다.
- 자식 프로세스가 부모 프로세스와 공유하고 있는 변수를 수정하여 새로운 물리 페이지를 할당 받는지 확인합니다.

```
$ test2  
[Test 2] Make a Copy  
[Test 2] pass
```

# test3.c

---

- fork() 시스템 콜을 통해 자식 프로세스를 생성합니다.
- 10개의 자식 프로세스를 생성하고, 각각의 자식 프로세스는 test2와 유사하게 부모 프로세스와 공유하는 변수를 수정합니다.
- 자식 프로세스의 종료 이후, free page의 적절한 회수가 이루어졌는지 확인합니다.

```
$ test3
[Test 3] Make Copies
child [0]'s result: 1
child [1]'s result: 1
child [2]'s result: 1
child [3]'s result: 1
child [4]'s result: 1
child [5]'s result: 1
child [6]'s result: 1
child [7]'s result: 1
child [8]'s result: 1
child [9]'s result: 1
[Test 3] pass
```

**Thank you**

