IGRUS Winter Bootcamp: Homework #6

Due on 2026.01.31

IGRUS

Contents

Chapter 02. Introduction to Operating Systems

Problem 1

프로그램이 실행되는 과정을 설명해 주세요. *(Fetch, Decode, Execute)*

Problem 2

시스템 콜(System call)이 무엇인지 설명해 주세요.

Problem 3

CPU 가상화에 대해 설명해 주세요. CPU 가상화는 어떤 "환상(illusion)"을 만들 수 있나요?

Problem 4

메모리 가상화에 대해 설명해 주세요. 메모리 가상화는 어떤 "환상(illusion)"을 만들 수 있나요?

Problem 5

프로세스의 주소 공간(address space) 개념과 물리 메모리(physical memory) 개념을 비교해 주세요. 일반적으로 한 프로세스 내의 메모리 참조는 다른 프로세스의 주소 공간에 영향을 주지 않는데, 이 이유도 설명해 주세요.

Problem 6

동시성 문제(The problem of Concurrency)에 대해 설명해 주세요.

Problem 7

다음 프로그램을 실행하면 동시성 문제가 발생할 수 있습니다. 동시성 문제가 발생하는 곳을 찾고, 이유를 설명해 주세요.

```c
#include <stdio.h>
#include <stdlib.h>
#include "common.h"

volatile int counter = 0;
int loops;

void* worker(void* arg) {
    int i;
    for (i = 0; i < loops; i++) {
        counter++;
```

```
        }
        return NULL;
    }

    int main(int argc, char* argv[]) {
        if (argc != 2) {
            fprintf(stderr, "usage: threads <value>\n");
            exit(1);
        }

        loops = atoi(argv[1]);
        pthread_t p1, p2;
        printf("Initial value : %d\n", counter);

        Pthread_create(&p1, NULL, worker, NULL);
        Pthread_create(&p2, NULL, worker, NULL);
        Pthread_join(p1, NULL);
        Pthread_join(p2, NULL);
        printf("Final value : %d\n", counter);

        return 0;
    }

```

Chapter 04. The Abstraction: The Process

Problem 8

CPU 가상화 기법 중 시분할(Time sharing) 기법에 대해 간단하게 설명해 주세요.

Problem 9

프로세스(Process)의 정의와 구성 요소에 대해 설명해 주세요.

Problem 10

프로세스와 관련된 특별한 레지스터에 대해 설명해 주세요.

- 프로그램 카운터(program counter, PC)
- 명령어 포인터(instruction pointer, IP)
- 스택 포인터(stack pointer), 프레임 포인터(frame pointer)

Problem 11

프로세스와 관련된 API는 크게 5 종류로 나눌 수 있습니다. 각 API에 대해 설명해 주세요. *(Create, Destroy, Wait, Miscellaneous Control, Status)*

Problem 12

프로세스의 생성 과정을 자세히 설명해 주세요. 프로세스의 메모리 구조(code, static data, heap, stack 영역), 데이터의 저장 위치를 포함해 설명해 주세요.

Problem 13

프로세스가 가질 수 있는 3가지 상태에 대해 설명해 주세요.

Problem 14

PCB(Process Control Block)에 대해 설명해 주세요.

Problem 15

Process list에 대해 설명해 주세요.

Chapter 05. Process API

Problem 16

`fork()` 시스템 콜에 대해 설명해 주세요. 부모 프로세스와 자식 프로세스의 메모리 공간, `fork()` 반환 값, `fork()` 호출 이후 동작 등을 비교해 주세요.

Problem 17

`wait()` 시스템 콜에 대해 설명해 주세요. 무엇을 강제하기 위해 사용하나요?

Problem 18

`exec()` 시스템 콜에 대해 설명해 주세요. `fork()` 시스템 콜과 비교하는 내용도 포함해 주세요.

Problem 19

`fork()`와 `exec()`은 일반적으로 함께 사용됩니다. 그렇다면 `fork()` 시스템 콜과 `exec()` 시스템 콜을 하나로 합치면 좋을 것 같은데, 왜 둘을 분리했을까요? 두 시스템 콜이 분리되어 있는 이유를 설명해 주세요.

Problem 20

Pipe 와 IO redirection 을 비교해 주세요.

Problem 21

Unix pipe 의 동작 원리를 설명해 주세요. 파일 디스크립터(File descriptor)를 포함하여 설명해 주세요.

Problem 22

`pipe()`, `dup()` 시스템 콜에 대해 설명해 주세요. *(주의: **`pipe()`** 시스템 콜과 Unix 의 pipe 는 서로 다른 것임)*