2021270660 이지원 - 운영체제 실습 과제 2
Process Counter

System Description

```
// 출력 포맷 지정을 위한 라이브러리
#include <iomanip>

// EnumProcesses()를 사용하기 위한 헤더 파일, 라이브러리 링크
#define PSAPI_VERSION 1 // OS 문제 해결을 위한 옵션 define ...
#include <windows.h>
#include <psapi.h>
#pragma comment( lib, "Psapi.lib" ) // #pragma comment(lib, "라이브러리.lib") -> 명시적 라이브러리 링크
```

프로세스 개수를 가져오기 위해 psapi.h 라이브러리를 불러왔습니다.

이외에도 상단에 생략된 오류 방지 코드와 표준입출력 라이브러리를 불러왔습니다.

시간 출력 함수와 프로세스 개수 출력 함수를 분리하여 모듈화하였고,

main() 함수에서 호출하여 사용합니다.

System Description

```cpp
// 현재 시간을 출력하는 함수
void printTime() {
    SYSTEMTIME lt;
    // GetSystem() -> UTC 기준... 따라서 GetLocalTime()을 사용해 시간을 가져옴
    GetLocalTime(&lt); // get local time
    cout << lt.wYear << "-";
    cout.width(2);
    cout.fill('0');
    cout << lt.wMonth << "-";
    cout.width(2);
    cout << lt.wDay << "  ";
    cout.width(2);
    cout << lt.wHour << ":";
    cout.width(2);
    cout << lt.wMinute << ":";
    cout.width(2);
    cout << lt.wSecond << " -> ";
}
```

GetSystem()은 UTC 기준의 시간을 출력하기 때문에 GetLocalTime()을 사용하였습니다.

SYSTEMTIME lt 변수를 참조 변수로 받아 할당한 뒤 wYear, wMonth, … 등으로 날짜와 시간을 각각 출력하였습니다.

System Description

```
// 현재 프로세스 수를 반환하는 함수
int getProcessNum() {
    DWORD aProcesses[1024];
    DWORD cbNeeded;

    // procesees가 없을 경우 return 1 ( 오류 )
    if (!EnumProcesses(aProcesses, sizeof(aProcesses), &cbNeeded)) {
        return 1;
    }

    return cbNeeded / sizeof(DWORD); // 프로세스의 개수 반환
}
```

EnumProcesses() 함수는 pid 리스트를 담을 배열과 그 배열의 사이즈를 전달 받아,

그리고 참조 변수를 통해 aProcesses 배열의 byte수를 return합니다.

이를 DWORD의 사이즈로 나누어 개수를 구할 수 있습니다.

System Description

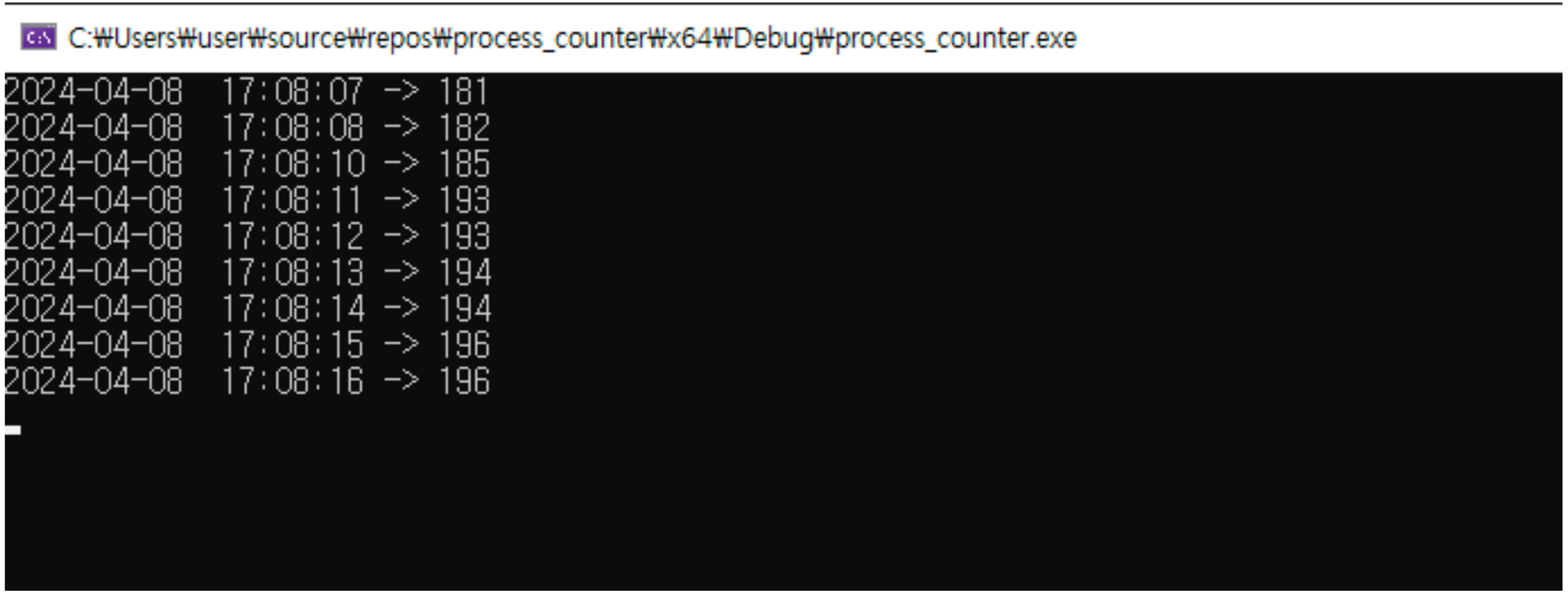
```
// 현재 프로세스 수를 반환하는 함수
int getProcessNum() {
    DWORD aProcesses[1024];
    DWORD cbNeeded;

    // procesees가 없을 경우 return 1 ( 오류 )
    if (!EnumProcesses(aProcesses, sizeof(aProcesses), &cbNeeded)) {
        return 1;
    }

    return cbNeeded / sizeof(DWORD); // 프로세스의 개수 반환
}
```

위와 같이 main() 함수에서 Sleep(1000)을 통해 1초 간격으로 함수를 호출하여 사용합니다.

Test Description



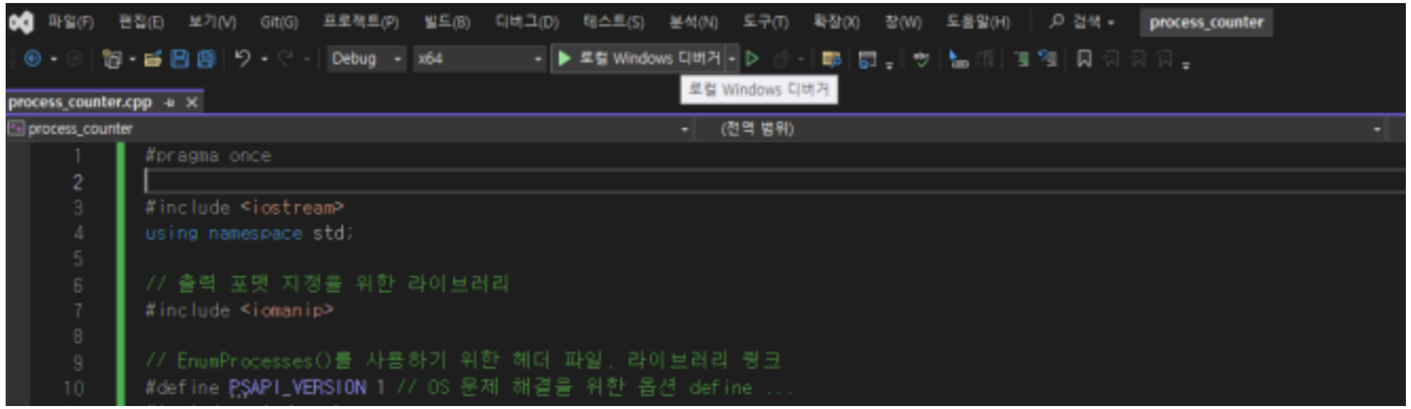
과제 예시와 다르게 가독성을 위해 :을 ->로 변경하였습니다.

1초 간격으로 프로세스 수가 정상적으로 출력되는 것을 확인할 수 있습니다.

User Documentation

1. .exe 파일 실행 (빌드된 상태)

2. 로컬 디버거로 실행 : 프로젝트 생성 후 소스 파일에 추가 후 로컬 디버거로 실행



3. 빌드 후 .exe 파일 실행 : 마찬가지로 프로젝트 생성 후 빌드 - 솔루션 빌드 후 .exe 파일을 찾아 실행

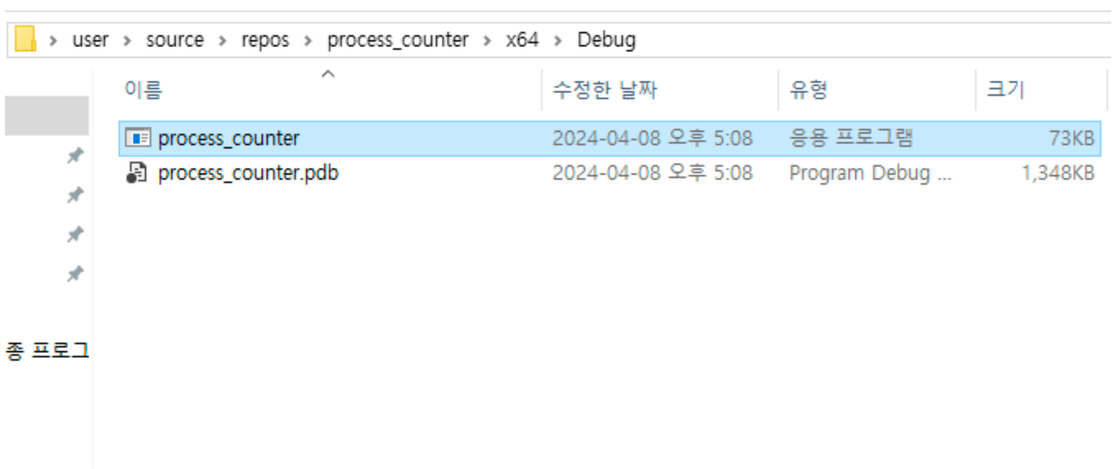


Table of contents

Self-evaluation

100/100

- 모듈화 적절히 사용하였음
- 주석 및 가독성 신경써서 작성
- 라이브러리 적절히 사용