

[HW#2] 다중 프로세스 소수 검색 프로그램 작성

* 할일

아래 조건을 만족하는 소수 구하기 프로그램을 작성

1. 프로그램 실행할 때 소수 검색 범위 종료값과 검색할 프로세스 개수를 argument로 지정
 - A. 예를 들어, 실행파일명이 primenumber일 때 “primenumber 1000 2”로 실행한 것은 1부터 1000까지 범위에서 2개의 프로세스로 소수를 검색함을 의미
2. 지정한 개수만큼 fork()로 자식 프로세스를 생성하고, 전체 소수 검색 범위를 자식 프로세스 개수만큼 균등하게 나누어 각 자식 프로세스는 담당 범위내에서 소수를 검색
 - A. 예를 들어, 검색 범위 최대값이 1000이고 프로세스 개수가 2이면, 자식 프로세스가 담당하는 검색 범위는 각각 1 ~ 500, 501 ~ 1000이다.
 - B. 검색 범위 최대값이 1000이고 프로세스 개수가 3이면, 자식 프로세스가 담당하는 검색 범위는 각각 1 ~ 333, 334 ~ 666, 667 ~ 1000이다.
3. 각 자식 프로세스는 자신의 프로세스 아이디, 담당 검색 범위, 찾은 소수의 개수를 출력한다.
 - A. 예를 들어, “Process[YYY] has found XX prime numbers in (SS ~ EE).”와 같이 출력한다. (YYY는 프로세스 아이디, XX는 소수의 개수, SS 검색 시작값, EE는 검색 종료값)
4. 각 자식 프로세스는 종료하면서 성공의 의미로 1을 리턴한다.
5. 부모 프로세스는 wait()로 모든 자식 프로세스들이 종료되길 기다린 후, 각 자식 프로세스가 보내는 리턴 값의 합을 구하여 마지막에 출력한다.
 - A. 예를 들어, “XX processes have completed their tasks.”와 같이 출력한다. (XX는 자식 프로세스가 보낸 리턴 값의 합)
6. 임의의 수 n이 소수인지 여부를 판별하는 함수는 아래 함수를 이용한다.

```
int isPrimeNumber(int n)
{
    int i = 0;
    int last = n/2;
    if(n <= 1)
        return 0;

    for(i = 2; i <= last; i++)
    {
        if( (n%i) == 0 )
            return 0;
    }
    return 1;
}
```

* 제출물

아래 두 파일을 zip으로 압축하여 하나의 압축 파일로 제출

1. 작성한 소스파일 (소스파일명은 hw2_본인학번.c 로 함)
2. 보고서 (아래 사항만 포함하고 보고서 파일명은 hw2_본인학번으로 함)
 - A. 작성한 코드에 대한 설명 기술
 - B. 소수 검색범위는 10, 100, 1000, 10000으로 변경하며, 각 소수 범위에 대해 프로세스 개수도 1 ~ 4까지 변경하며 각 실행결과 캡처화면