System Programming

Assignment#2 _ Finding Frequency Distribution

컴퓨터공학과

201811259

배수빈

Multi-process program , Multi-thread program

-각 구조에서 처리하는 숫자의 갯수 :

Int eachProcess, eachThread (자식프로세스, 생성된쓰레드가 처리하는 숫자갯수)

= amout (전체숫자의 개수) / n (자식프로세스, 생성된쓰레드의 수)

Int remainder (부모프로세스, 쓰레드의 메인에서 처리하는 숫자 개수)

= amount – eachProcess(eachThread) * n

(ex. Dataset숫자는 10개, 자식프로세스 or 생성되는쓰레드 3개

➔ 자식프로세스, 쓰레드함수에서는 숫자3개,

부모프로세스, 메인에서는 숫자1개를 처리한다.)

단, 입력받은 프로세스의 수가 1일경우 부모프로세스, 메인에서 모든 숫자를 처리하므로 eachProcess, eachThread = 0 (fork, pthread_create 하지 않음)

-전체적인 흐름

멀티프로세스 : 메인에서 입력받은 프로세스개수(첫번째 인자값) – 1 개의 프로세스를 fork한뒤 각 자식 프로세스는 병렬로 eachProcess개의 숫자들을 처리하고, 메시지큐로 도수분포배열의 인덱스와 그인덱스의 값을 send한다. 부모프로세스에서 메시지큐에 값이 들어오는대로 receive하고 부모프로세스의 도수분포배열의 해당 인덱스에 값을 더한다. 이후 remainder개의 숫자를 마저 처리한뒤 최종 도수분포배열을 출력한다.

멀티쓰레드 : 메인에서 입력받은 쓰레드개수(첫번째 인자값) -1 개의 쓰레드를 create하고 이렇게 생성된 쓰레드는 병렬로 쓰레드함수인 thread_function을 실행한다. 이 thread_function에서는 eachThread개의 숫자들을 처리하고 모든 쓰레드와 메인이 참조할수 있는 전역변수인 도수분포배열의 값들을 바로바로 증가시킨다. 메인에서는 remainder개의 숫자를 처리한뒤 마찬가지로 도수분포배열의 값들을 증가시킨다. 그리고 최종 배열을 출력한다.

-숫자를 '처리'하는 방법

우선 최종으로 출력하는 배열인 returnRank은 [0~ i-1][i ~ 2i-1]...[@i ~ 9999]의 형태를 가진다. 만약 i가 1000이면 900이라는 숫자는 0~9999의 자리에 해당하는 것이다.

```
void dosuAlgorithm(int* returnRankArr[], int i, int inputNum) {
        int rank = inputNum / i;
        returnRankArr[rank]++;
}
```

멀티프로세스, 멀티쓰레드에서 모두 내부알고리즘이 같은 함수를 사용한다. (인자값의 차이만 있다.) 알고리즘은 처리할 숫자를 interval 즉 i로 나눈뒤 returnRank에서 그 몫을 인덱스로 가지는 자리의 값을 1 증가시키는것이다.

멀티프로세스에서 returnRank는 fork이전에 선언을 해주므로 모든 프로세스가 같은 형태의 배열을 갖는다. send할 때 값과 인덱스를 넘겨주고 receive하면서 부모프로세스의 returnRank의 해당인덱스에 값을 더해준다.

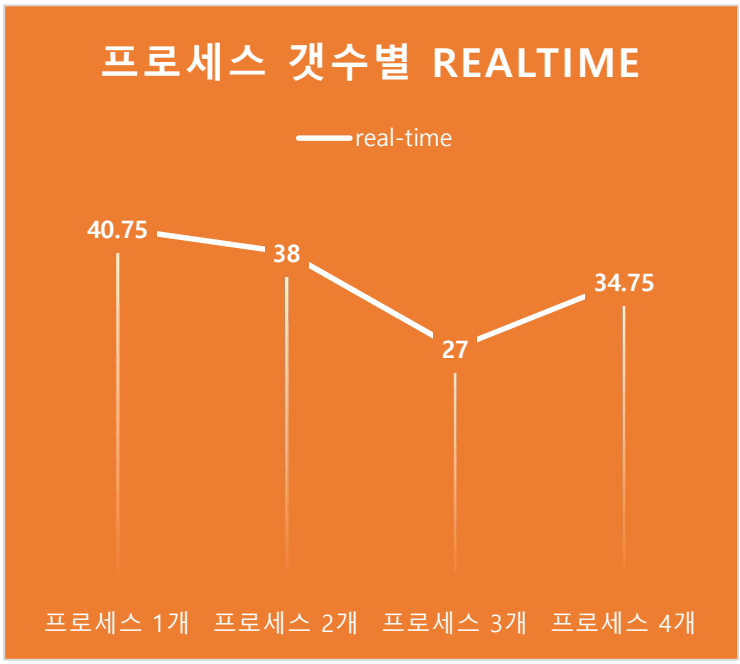
멀티쓰레드에서는 위의 첨부한 사진과 같이 배열포인터를 넘겨받는다. returnRank배열이 모든 쓰레드에서 참조가 가능해야 하기 때문에 포인터배열로 선언해둔 뒤 메인에서 interval에 따라 계산을 한 뒤 필요한 만큼 동적할당을 해주기 때문이다.

-Multi의 효과 (time분석)

1) Multi-process

$ time ./ku_pfred n 1000 dataset 의 real값 (dataset의 숫자 : 9999개)

시행횟수 / n	1회	2회	3회	4회	평균값
1개	46s	38s	40s	39s	40.75s
2개	41s	37s	38s	36s	38s
3개	23s	26s	35s	24s	27s
4개	33s	36s	28s	42s	34.75

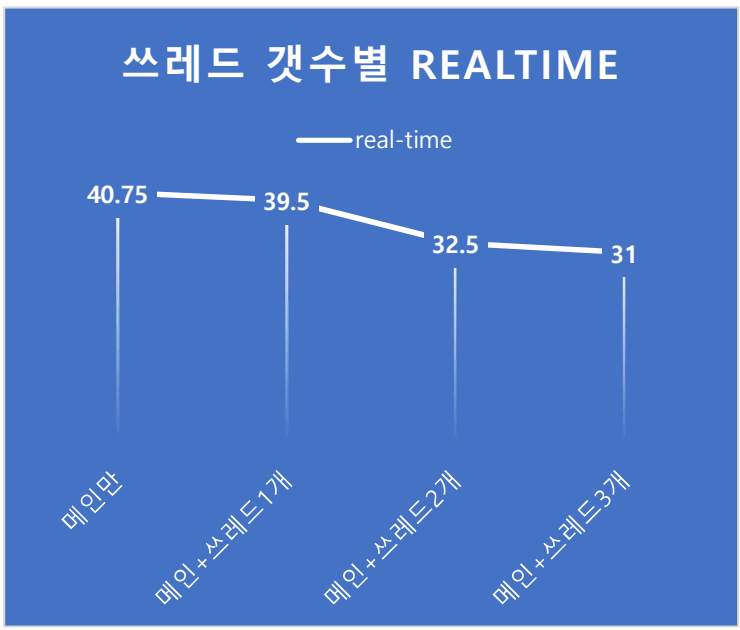


N이 1일때는 부모프로세스에서, N이 2일때는 1개의 자식프로세스에서 모든 숫자를 처리한다. 따라서 시간이 많이 걸리고 프로세스가 늘어남에 따라 병렬적으로 일을 나눠서 할 수 있기 때문에 시간이 줄어든다. 프로세스가 3개가 되고서야 자식프로세스2개가 병렬로 일을 하기 때문에 시간이 11초나 감소한다.

2) Multi-thread

$ time ./ku_tfred n 1000 dataset 의 real값 (dataset의 숫자 : 9999개)

시행횟수 / n	1회	2회	3회	4회	평균값
1개	47s	37s	43s	40s	41.75s
2개	37s	41s	44s	36s	39.5s
3개	28s	35s	31s	36s	32.5s
4개	32s	37s	26s	29s	31s



n이 1일 때는 메인에서, n이 2일때는 create되는 1개의 쓰레드에서 모든 숫자를 처리한다. 따라서 시간이 가장 많이 걸리고 쓰레드가 늘어남에 따라 각 쓰레드들이 숫자를 나누어 적은 숫자들을 병렬적으로 처리하기 때문에 시간이 줄어든다. 점점 시간이 줄어드는 이유도 처리하는 숫자가 줄어들기 때문이다.

-함수

Function Name	Arguments	Description
void dosuAlgorithm	int returnRank[] int * returnRank[]	함수내부 계산으로 얻은 인덱스의 값을 ++해줄 배열의 주솟값을 넘겨받는다.
	int i	입력받은 도수분포표의 interval을 넘겨받는다.
	int inputNum	도수분포 처리를 할 숫자를 넘겨받는다.
void fileRead	char * filename	오픈할 파일의 이름을 넘겨받는다.
	int readBytes	파일내부에서 읽을 바이트수를 넘겨받는다.
	int Offset	파일내부에서 어디서부터 읽을지 읽는 부분의 offset을 넘겨받는다.
	char retore[]	Offset으로부터 readBytes만큼 읽은 글자를 저장할 char배열의 주소를 넘겨받는다.