시스템 프로그래밍

MPP, MTP



컴퓨터공학과

201411292 이민환

Multi-process program

void file\_open(char \*inputname)

커맨드를 통해 input\_file을 입력 받아 텍스트를 열어 텍스트 안의 수를 입력 받은 m만큼 배열에 차례로 넣어준다.

void make\_process()

n의 값을 커맨드를 통해 입력 받아 n값 만큼 for문을 통해 fork를 하여 프로세스를 생성한다. 이 때, pid = 0이면, break를 걸어 프로세스가 1개씩 생성되도록 하였다. 그리고 프로세스를 생성할 때 마다 p\_num을 1씩 크게 해주어 프로세스를 구분할 수 있게 했다.

void process\_sort()

for문을 통해 프로세스 생성이 끝나면, pid = 0인 프로세스들에게 텍스트에서 수를 받아와 저장 했던 배열의 수들을 알맞게 분배해준다. 이 때, m / n을 하여 몫들을 프로세스에 나눠주고 나머지는 마지막 프로세스에 다 넣어준다. arr\_copy와 quick\_sort 를 다 하고 난 후 mq\_send를 통해 정렬된 함수들을 우선순위에 따라 보내준다. 이 때, 우선순위는 p\_num을 통해 차례로 들어가게 해준다.

void arr\_copy(int p\_arr[], int num[])

프로세스에 분배할 수에 맞춰 p\_arr 배열을 동적 할당 해주어 수를 차례로 집어 넣는다.

void quick\_sort(int left, int right)

p\_arr 배열에 다 분배를 해준 후 quick\_sort를 통해 정렬 해준다. 재귀를 통해 계속 정렬하여 결국, 배열의 모든 수가 오름차순으로 정렬이 된다.

void mq\_receiver()

pid != 0, 즉 부모 프로세스는 자식 프로세스들이 sorting을 할 때 까지 wait하다가 메시지를 다 보내고 종료를 했을 때 mq\_receiver를 통해 메시지를 받아 배열에 그대로 저장을 해준다.

void file\_write(char \*outputname)

sorting이 끝난 후 커맨드에서 받아온 output\_file을 인자값으로 받아와 텍스트를 열어 배열의 값을 순서대로 적어주고 저장한다.

Multi-thread program

void file\_open(char \*inputname)

커맨드를 통해 input\_file을 입력 받아 텍스트를 열어 텍스트 안의 수를 입력 받은 m만큼 배열에 차례로 넣어준다.

void make\_thread(pthread\_t \*p\_thread)

n의 값을 커맨드를 통해 입력 받아 n값 만큼 for문을 통해 create를 하여 쓰레드를 생성한다. thread\_sort 함수를 실행시켜 주는데 이 때, 인자 값으로 쓰레드마다 1, 2, 3, 4 순으로 가져가 쓰레드를 구분해주기 위해 필요한 변수를 가져간다.

void thread\_sort(void \*temp)

쓰레드마다 텍스트를 통해 만든 배열에 sorting 범위를 정해 주기 위해 인자 값으로 받아온 스레드 번호를 이용하여 배열의 각 시작 위치를 지정해준다. 각 쓰레드마다 (temp-1)\*(m/n)을 하여 시작 위치를 지정하고, 마지막 위치의 경우에는 시작위치에 m/n을 더해주고 -1을 한다. 마지막 쓰레드의 경우 나머지 값들을 다 가져가기 위해 마지막 위치를 m-1로 지정 해준다.

void quick\_sort(int left, int right)

쓰레드에 수를 다 지정해 주었으면 quick\_sort를 통해 정렬을 해준다. 재귀를 통해 계속 정렬하여 결국, 배열의 모든 수가 오름차순으로 정렬이 된다.

void file\_write(char \*outputname)

sorting이 끝난 후 커맨드에서 받아온 output\_file을 인자값으로 받아와 텍스트를 열어 배열의 값을 순서대로 적어주고 저장한다.