

1. Sorting Number by Multi-Process program

Function Name	Arguments	Description
main	int argc	실행 시 입력한 인자 수
	char ** argv	argv[1] : 정렬할 숫자의 수 argv[2] : 프로세스 수 argv[3] : 숫자들이 저장된 파일 argv[4] : 결과를 저장할 파일
	Return Value	X
Function Description	전체 프로그램을 총괄하며, 정렬할 숫자의 수와 그 수들이 저장된 파일을 읽어 정렬되지 않은 숫자들을 불러온다. 이후 프로세스 수에 맞게 프로세스를 생성한 후 적절하게 숫자의 양을 분할하여 각 프로세스에 메시지 큐를 이용해 전달하여 정렬하도록 한다.. 정렬된 숫자는 메시지 큐를 이용해 전달받아 마무리 정렬 후 파일을 저장한 후 종료한다.	

Function Name	Arguments	Description
sender_main	SENDPOINT * sendpoint	typedef struct sendPoint { int start; int end; }SENDPOINT;
	Return Value	X
Function Description	Main에서 보낸 숫자들을 Message Queue로 전송하는 역할을 한다. Main에서 반복문을 이용해 호출하므로, 프로세스의 개수만큼 이 함수가 호출된다.	

Function Name	Arguments	Description
recv_process		.
	Return Value	X
Function Description	메인에서 전송한 숫자들을 메시지큐를 이용해 전달받아 이들을 정렬하는 역할을 한다. 정렬 후 다시 메시지큐를 이용해 메인에서 활용하도록 반환한다.	

Function Name	Arguments	Description
recv_master	Int p_count	
	Return Value	X
Function Description	각 프로세스에서 정렬한 숫자값을 메시지큐를 이용해 받아 오는 역할을 한다. 프로세스 개수만큼 메시지큐를 이용해 받기 때문에 프로세스의 개수를 인자로 받아 그 횟수만큼 반복한다.	

2. Sorting Number by Multi-Thread program

Function Name	Arguments	Description
Sort_function	void* data	분석할
	Return Value	X
Function Description	자신의 고유한 시작점을 data값으로 받아와 그 부분부터 버블정렬을 진행한다. 버블정렬을 하기위해 두 숫자를 비교해야하므로 그 두 숫자 비교에 대한 뮤텍스를 각각 가지고있다. 또한 모든 쓰레드들이 카운트값을 공유하여 두 수의 위치를 바꾸지 않아도 되는 횟수가 일정 이상 되면 쓰레드를 종료하도록 한다.	

Function Name	Arguments	Description
Main	int argc	실행 시 입력한 인자 수
	char ** argv.	argv[0] : 정렬할 숫자의 수
		argv[1] : Thread 수
		argv[2] : 숫자가 저장된 파일 argv[3] : 결과를 저장할 파일
	Return Value	X
Function Description	전체 프로그램을 총괄하며, 정렬할 숫자의 수와 그 수들이 저장된 파일을 읽어 정렬되지 않은 숫자들을 불러온다. 이후 프로세스 수에 맞게 쓰레드를 생성한 후 적절하게 숫자의 양을 분할하여 각 쓰레드에 범위 인덱스값을 인자로 전달하여 정렬하도록 한다. 모든 쓰레드가 종료되기를 기다렸다가 정렬된 숫자를 저장하고 종료한다.	