

# 2021년 1학기

## 리눅스시스템프로그래밍 설계과제 명세서

### 세포 매트릭스 게임 구현

#### 게임 진행 규칙

- $(m \times n)$  크기의 2차원 매트릭스를 입력으로 받는다.
- 입력받은 2차원 매트릭스의 한 칸은 하나의 세포라고 가정한다.
- 하나의 세포에 이웃하는 8칸은 이웃 세포라고 가정한다.
- 세포 칸에 값 1이 들어있으면 살아있는 세포, 0이 들어있으면 죽은 세포라고 가정한다.  
(단, 세포 칸의 값은 0과 1 이외의 값은 갖지 않는다)
- 가장자리 세포들의 경우 매트릭스 밖의 이웃 세포는 모두 0이라고 가정한다.
- 임의의 값으로 설정된 "input.matrix" 파일로 입력되는 2차원 매트릭스를 0세대라 한 후, 각각의 모든 세포에 대해 아래의 규칙을 적용한 후 나오는 결과를 다음 세대라고 한다.  
(예: 0세대 다음은 1세대, 4세대 다음은 5세대)
  1. 해당 세포가 살아있을 때, 살아있는 이웃 세포의 수가 2개 이하 혹은 7개 이상인 경우 해당 세포는 다음 세대에 죽는다.
  2. 해당 세포가 살아있을 때, 살아있는 이웃 세포의 수가 3개 ~ 6개인 경우 해당 세포는 다음 세대에 살아서 유지된다.
  3. 해당 세포가 죽어있을 때, 살아있는 이웃 세포의 수가 4개인 경우 해당 세포는 다음 세대에 살아난다.
  4. 해당 세포가 죽어있을 때, 살아있는 이웃 세포의 수가 4개를 제외한 나머지 경우에는 해당 세포는 다음 세대에 여전히 죽어있는 상태가 유지된다.

#### 구현 요구사항

- 2차원  $(m \times n)$  매트릭스는 내부 세포들의 초기 값이 0 또는 1로 무작위 배정된 초기 파일을 (input.matrix) 입력으로 받아 게임을 시작한다. (단, 매트릭스의 크기인  $m, n$ 은 1 이상의 자연수이며 최대값은 10,000으로 가정한다)
- 파일 입출력을 통해 초기 파일 (input.matrix) 읽기, 세대별 중간 결과 파일 (gen\_n.matrix) 생성, 최종 결과 파일 (output.matrix) 생성 작업을 진행한다.
- 하나의 프로세서로 (하나의 스레드) 병렬처리 없이 진행하는 버전과 (20점) 두 가지 병렬처리 버전 (child process 생성을 통한 병렬처리, thread 생성을 통한 병렬처리 각 40점), 총 세 가지 버전을 구현해야 한다.
- 프로그램은 입력 파일을 (input.matrix) 인자로 받아서 실행을 시작한다. 프로그램이 실행되면, 첫 번째로 사용자로부터 어떤 방식으로 동작을 수행할 것인지 입력받는다.  
(1) 프로그램 종료 (2) 순차처리 (3) Process 병렬처리 (4) Thread 병렬처리 중 하나의 번호를 입력 받아, 해당 방식으로 게임 동작을 처리한다.
- 전 단계에서 (1), (2) 가 선택되면 바로 다음 단계로 넘어가고, (3) 또는 (4)가 선택되면 사용자로부터 생성할 Child Process 또는 Thread 개수를 입력받는다. 하나의 세대 진행에 필요한 매트릭스 세포별

계산은 사용자로부터 입력받는 임의의 자연수 개수의 process 또는 thread를 생성하여 병렬 처리를 수행한다. 생성된 process/thread는 입력 매트릭스 행의 크기를 기준으로 최대한 균등하게 계산을 나누어서 수행해야 한다. 이때, 부모 프로세스와 메인 thread는 별도의 매트릭스 계산을 수행하지는 않는다.

- 최소 1 이상의 세대 수를 (최대는 int 변수로 표현 가능한 수) 사용자로부터 입력받은 후 게임을 진행한다. 게임 종료 (각 세대 진행에 따른 중간 파일들 및 최종 결과 파일(output.matrix) 생성) 후에는 게임 진행을 위해 사용된 프로세서들의 ID (Process 병렬처리 경우), 스레드들의 ID (Thread 병렬처리 경우), 총 수행 시간 (입력 파일 읽기부터 최종 결과 파일 생성까지의 시간)을 ms 단위로 출력한 후에, 다시 어떤 방식으로 동작을 수행할 것인지를 입력받는 단계로 돌아간다.

## 제출 유의사항

- 결과물로는 보고서와 소스코드를 제출해야 하며, 소스코드가 다수의 파일로 구성되었을 경우에는(다중 모듈 프로그래밍) Makefile을 함께 제출해야 한다. 실행 파일은 제출하지 않음. (\* 소스코드가 하나의 파일인 경우에는 "gcc"로 실행 파일을 만들고, 소스코드가 여러 개인 경우에는 "make"를 수행하여 실행 파일을 생성하여 테스트를 진행할 예정임)
- 임의로 생성된 입력 파일을 사용하여 임의의 동작 방식, 임의의 프로세스/스레드 개수, 임의의 세대 수에 대해 올바르게 최종 파일이 생성되는지 여부와 3가지 구현 방식 (순차처리, process 생성을 통한 병렬처리, thread 생성을 통한 병렬처리)에 대해 사용된 프로세스/스레드 ID, 총 수행 시간 등의 결과가 정상적으로 출력되는지 확인할 예정임
- 보고서는 워드(한글, MS-Word, PDF) 파일을 제출해야 함. 표지는 만들지 말고, 첫 장, 첫 줄에 학과, 학번, 성명 정보를 오른쪽 정렬로 기입하고, 바로 밑에 보고서를 작성함. 구현한 프로그램에 대해 1. 구현 개요 (구현을 완료한 기능에 대해서 보고서에 상세하게 기술해야 함), 2. 소스코드, 3. 실행 결과 (캡처), 4. 구현을 통해 배운 점 (입력 크기, 동작 방식, 세대 수에 따른 수행 시간 결과를 보고 배운 점 포함)을 작성해야 함. 문서 스타일은 자유 양식. 단, 보고서 본문 글자 크기는 10pt 사용
- 보고서와 소스코드는 각각 하나의 파일로 제출해야 함. 보고서는 제출 시 "#학번\_성명.hwp" (예. #20191234\_홍길동.hwp)로 파일 이름을 명명해야 함. 소스코드는 하나의 파일로 구현한 경우에는 "학번.c" (예. 20191234.c)로 제출하고, 여러 개의 파일로 구현한 경우에는 Makefile까지 포함하여 tar로 묶어서 "학번.tar" (예. 20191234.tar) 파일을 제출해야 함
- 스마트캠퍼스LMS를 통해 제출
- **제출기한: 2021년 6월 6일 오후 11:59:59 (기한 후 제출 불가)**