



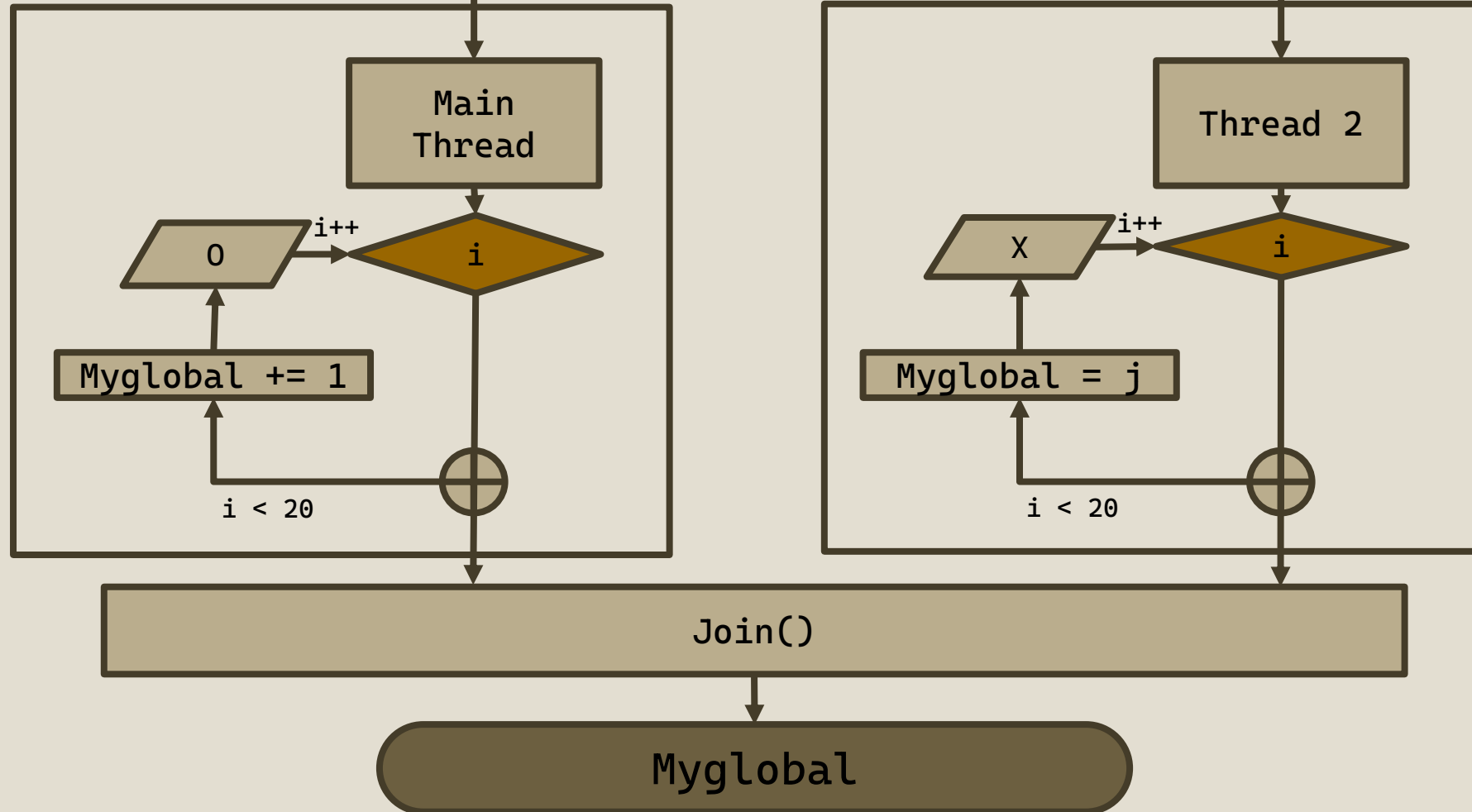
THREAD2.C THREAD3.C

2019305061 임주형

Thread2.c

- POSIX 스레드 (pthread) 라이브러리를 활용해 스레드를 생성
- 생성한 스레드와 메인 스레드가 동시에 전역변수 값을 증가시키며 출력

thread2.c



Thread2.c 코드 및 결과

thread2.c

```
1 #include <pthread.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4 #include <stdio.h>
5
6 int myglobal;
7 void *thread_function(void *arg){
8     int i, j;
9     for(i = 0; i < 20; i++){
10         j = myglobal;
11         j = j + 1;
12         printf("x");
13         fflush(stdout);
14         sleep(1);
15         myglobal = j;
16     }
17     return NULL;
18 }
```

```
~$ cc -D_REENTRANT thread2.c -o thread2 -lpthread
```

```
~$ ./thread2
```

```
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

```
myglobal equals 21
```

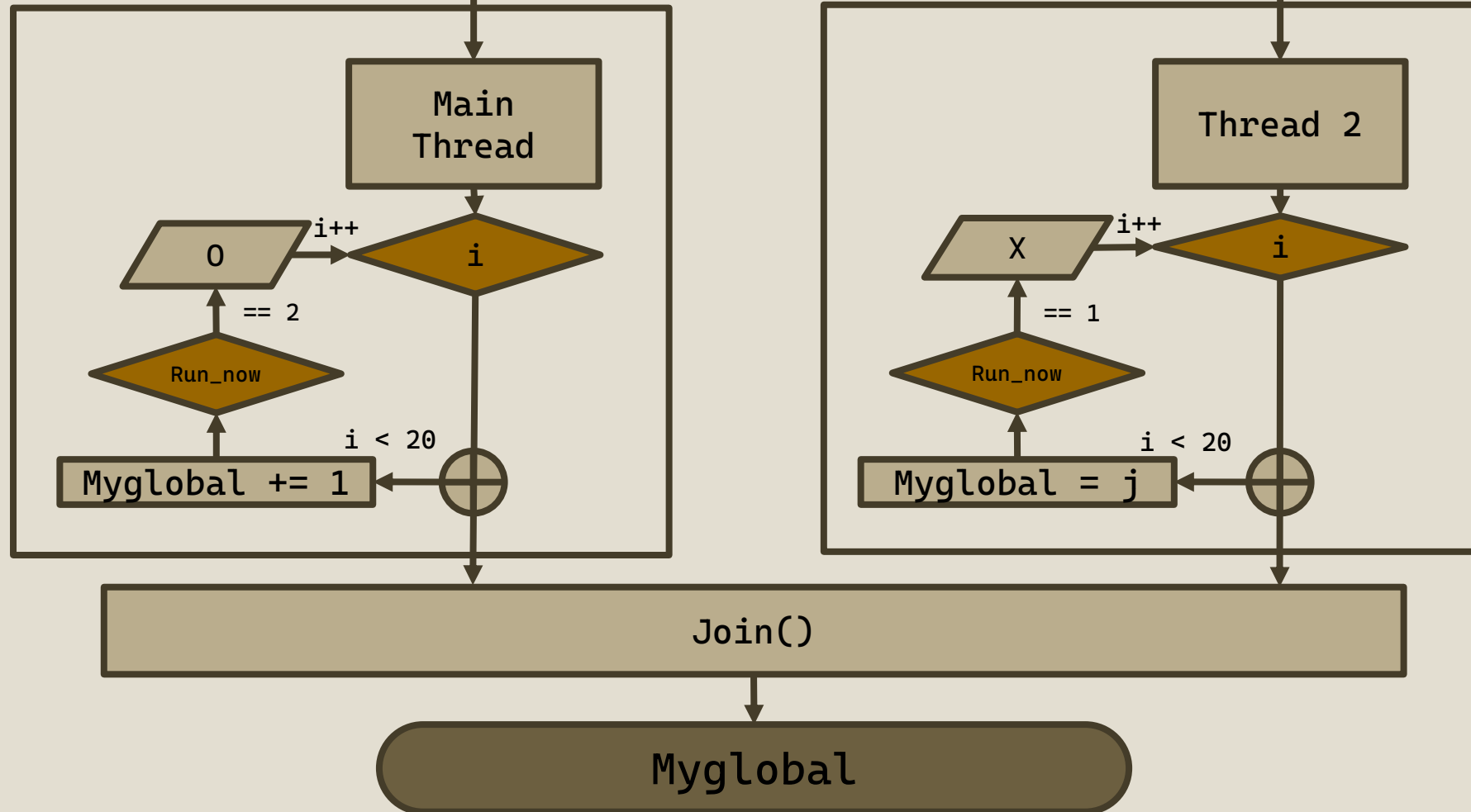
```
~$ █
```

```
20 int main(void){
21     pthread_t mythread;
22     int i;
23
24     if(pthread_create (&mythread, NULL, thread_function, NULL)){
25         printf("ldquo; error creating thread.");
26         abort();
27     }
28
29     for(i = 0; i < 20; i++){
30         myglobal = myglobal + 1;
31         printf("o");
32         fflush(stdout);
33         sleep(1);
34     }
35
36     if(pthread_join (mythread, NULL)){
37         printf("error joining thread.");
38         abort();
39     }
40     printf("#myglobal equals %d#n", myglobal);
41     exit(0);
42 }
```

Thread3.c

- POSIX 스레드 (pthread) 라이브러리를 활용해 스레드를 생성
- 생성한 스레드와 메인 스레드가 동시에 전역변수 값을 증가시키며 출력
- Run_now 전역변수의 값에 따라 각 스레드의 출력 조정

thread3.c



Thread3.c 코드 및 결과

 Thread3.c

```

1 #include <pthread.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4 #include <stdio.h>
5
6 int myglobal;
7 int run_now = 1;
8
9 void* thread_function(void* arg) {
10     int i, j;
11     for (i = 0; i < 20; i++) {
12         j = myglobal;
13         j = j + 1;
14         if (run_now == 1) {
15             printf("x");
16             run_now = 2;
17         }
18         fflush(stdout);
19         sleep(1);
20         myglobal = j;
21     }
22     return NULL;
23 }

```

```

25 ▾ int main(void) {
26     → pthread_t mythread;
27     → int i;
28
29 ▾ → if (pthread_create(&mythread, NULL, thread_function, NULL)) {
30     → → printf("ldquo; error creating thread.");
31     → → abort();
32     → }
33
34 ▾ → for (i = 0; i < 20; i++) {
35     → → myglobal = myglobal + 1;
36 ▾ → → if (run_now == 2) {
37     → → → printf("o");
38     → → → run_now = 1;
39     → → }
40     → → fflush(stdout);
41     → → sleep(1);
42     → }
43
44 ▾ → if (pthread_join(mythread, NULL)) {
45     → → printf("error joining thread.");
46     → → abort();
47     → }
48     → printf("#myglobal equals %d\n", myglobal);
49     → exit(0);
50 }

```

[illegible]