

1. 2016-1

Circle or cross: "T" if True – "F" if False.

- T / F A semaphore is a data structure.
- T / F Semaphores can not be used for avoiding dead locks
- T / F A monitor is a programming language construct
- T / F Monitors encapsulate shared data structures.
- T / F Both semaphores and monitors are distributed as function calls.
- T / F Monitors use condition variables, while semaphores do not.

2. 2016-2

```
001 /*                                030 void* thread3 (void* a) {
002 * (c) 2015-2016 Rahmat M. Samik-Ibrahim 031     printf("T3X\n");
002 * -- This is free software              032     sem_post (&sem[6]);
003 * Feel free to copy and/or modify and/   033     sem_post (&sem[2]);
004 * or distribute it, provided this notice, 034 }
004 * and the copyright notice, are preserved. 035
005 * REV04 Tue Dec 13 15:19:04 WIB 2016     036 void* thread4 (void* a) {
006 * START Wed Sep 30 00:00:00 UTC 2015     037     sem_wait (&sem[4]);
007 */                                        038     printf("T44\n");
008                                           039     sem_wait (&sem[5]);
009 #include <stdio.h>                       040     printf("T45\n");
010 #include <stdlib.h>                       041     sem_wait (&sem[6]);
011 #include <semaphore.h>                   042     printf("T46\n");
012 #include "99-myutils.h"                 043 }
013 #define nSem 7                           044
014                                           045 void main(void) {
015 sem_t sem[nSem];                          046     printf("MAIN\n");
016                                           047     for (int ii=1;ii<nSem;ii++)
017 void* thread1 (void* a) {                 048         sem_init(&sem[ii], 0, 0);
018     sem_wait (&sem[1]);                    049     daftar_trit (thread1);
019     printf("T1X\n");                       050     daftar_trit (thread2);
020     sem_post (&sem[4]);                    051     daftar_trit (thread3);
021 }                                          052     daftar_trit (thread4);
022                                           053     jalankan_trit ();
023 void* thread2 (void* a) {                 054     beberes_trit ("T3EXIT");
024     sem_wait (&sem[2]);                      055 }
025     printf("T2X\n");
026     sem_post (&sem[5]);
027     sem_post (&sem[1]);
028 }
```

Write down the program output:

3. 2017-1

Program Code of Synchronization (using 99-myutils.h and 99-myutils.c from the lab assignment)

```

001 /* (c) 2011-2017 Rahmat M. Samik-Ibrahim
002 * This is free software. Feel free to copy and/or
003 * modify and/or distribute it, provided this
004 * notice, and the copyright notice, are preserved.
005 * REV01 Wed May 17 17:02:37 WIB 2017
006 * START Wed May 3 12:58:28 WIB 2017
007 *
008 * sem_init(), sem_wait(), sem_post(): semaphore
009 * sleep(X): sleep X seconds
010 * daftar_trit(T): register thread T
011 * jalankan_trit(): start all registered threads.
012 * beberes_trit(): exit all threads above. */
013 #define jmlKIRI 3
014 #define jmlKANAN 2
015 #define SLEEP 1
016 #include <stdio.h>
017 #include <stdlib.h>
018 #include <semaphore.h>
019 #include <unistd.h>
020 #include "99-myutils.h"
021 sem_t mutexID, syncModKiri, syncModKanan;
022 sem_t syncKiriMod, syncKananMod;
023 int sequence = 0;
024
025 void cetak(char* posisi) {
026     sem_wait (&mutexID);
027     printf("%s (%d)\n", posisi, sequence++);
028     fflush(NULL);
029     sem_post (&mutexID);
030     sleep(SLEEP);
031 }
032 void* Kanan (void* a) {
033     while (TRUE) {
034         sem_wait (&syncModKanan);
035         cetak("--++Kanan");
036         sem_post (&syncKananMod);
037     }
038 }
039 void* Kiri (void* a) {
040     while (TRUE) {
041         cetak("Kiri-+++-");
042         sem_post (&syncKiriMod);
043         sem_wait (&syncModKiri);
044     }
045 }
046 void* Moderator (void* a) {
047     int ii;
048     while (TRUE) {
049         for (ii=0; ii<jmlKIRI; ii++)
050             sem_wait (&syncKiriMod);
051         for (ii=0; ii<jmlKANAN; ii++)
052             sem_post (&syncModKanan);
053         for (ii=0; ii<jmlKANAN; ii++)
054             sem_wait (&syncKananMod);
055         for (ii=0; ii<jmlKIRI; ii++)
056             sem_post (&syncModKiri);
057     }
058 }
059 int main(int argc, char * argv[]) {
060     int ii;
061     sem_init (&syncModKiri, 0, 0);
062     sem_init (&syncModKanan, 0, 0);
063     sem_init (&syncKiriMod, 0, 0);
064     sem_init (&syncKananMod, 0, 0);
065     sem_init (&mutexID, 0, 1);
066
067     for (ii = 0 ; ii < jmlKANAN; ii++)
068         daftar_trit(Kanan);
069     for (ii = 0 ; ii < jmlKIRI; ii++)
070         daftar_trit(Kiri);
071     daftar_trit(Moderator);
072
073     jalankan_trit();
074     beberes_trit("Selese...");
075 }

```

Write down the next 5 lines of the program output:

K i r i - + - + - + - (0)
