

1. 2016-1

```
001 /* FORK
002  * (c) 2015-2016 M. Anwar Ma'sum and Rahmat M. Samik-Ibrahim
003  * This is a free software ----- Rev. 06 - 01-Apr-2016
004  */
005
006 #include <stdio.h>
007 #include <sys/types.h>
008 #include <unistd.h>
009
010 void main() {
011     pid_t  pid1, pid2, pid3;
012
013     pid1 = pid2 = pid3 = getpid();
014     printf(" 2016  2015  2014--\n=====\n");
015     printf("[%4d] [%4d] [%4d]\n", pid1, pid2, pid3);
016     fork();
017     pid1 = getpid();
018     wait(NULL);
019     pid2 = getpid();
020     if(!fork()) {
021         pid2 = getpid();
022         fork();
023     }
024     pid3 = getpid();
025     wait(NULL);
026     printf("[%4d] [%4d] [%4d]\n", pid1, pid2, pid3);
027 }
```

- (a) (KOLOM) Lingkari tahun angkatan anda berikut ini: (A) 2016 (B) 2015 (C) lainnya.
- (b) (BARIS) Lingkari sesuai angka terakhir (paling kanan) dari NPM anda: 0 1 2 3 4 5 6
- (c) Harap mengisi (KOLOM:BARIS) dengan 1000
- (d) Harap mengisi kolom dan baris lainnya sesuai dengan keluaran program di atas!

NPM	2016	2015	Lainnya
0	[]	[
1	[]	[
2	[]	[
3	[]	[
4	[]	[
5	[]	[
6	[]	[

3. 2017-1

Program Code of Processes and Threads	
001 /* 002 * (c) 2005-2017 Rahmat M. Samik-Ibrahim 003 * This is free software. Feel free to copy and/or 004 * modify and/or distribute it, provided this 005 * notice, and the copyright notice, are preserved. 006 * REV02 Wed May 17 16:52:02 WIB 2017 007 * REV00 Wed May 3 17:07:09 WIB 2017 008 * 009 * fflush(NULL): flushes all open output streams 010 * fork(): creates a new process by cloning 011 * getpid(): get PID (Process ID) 012 * wait(NULL): wait until the child is terminated 013 * 014 */ 015 016 #include <stdio.h> 017 #include <unistd.h> 018 #include <sys/types.h>	019 #include <sys/wait.h> 020 #include <stdlib.h> 021 022 void main(void) { 023 int firstPID = (int) getpid(); 024 int RelPID; 025 026 fork(); 027 wait(NULL); 028 fork(); 029 wait(NULL); 030 fork(); 031 wait(NULL); 032 033 RelPID=(int)getpid()-firstPID+1000; 034 printf("RelPID: %d\n", RelPID); 035 fflush(NULL); 036 }

Program Output (line 34 of every process):

R e l P I D :

4. (6 points) 2017-2

The Program Code	
001 /* 002 * (c) 2017 Rahmat M. Samik-Ibrahim 003 * http://rahmatm.samik-ibrahim.vlsm.org/ 004 * This is free software. 005 * REV02 Mon Dec 11 17:46:01 WIB 2017 006 * START Sun Dec 3 18:00:08 WIB 2017 007 */ 008 009 #include <stdio.h> 010 #include <unistd.h> 011 #include <sys/types.h> 012 #include <sys/wait.h> 013 014 #define LOOP 3 015 #define OFFSET 1000	017 void main(void) { 018 int basePID = getpid() - OFFSET; 019 020 for (int ii=0; ii < LOOP; ii++) { 021 if(!fork()) { 022 printf("PID[%d]-PPID[%d]\n", 023 getpid() - basePID, 024 getppid() - basePID); 025 fflush(NULL); 026 } 027 } 028 }

Program Output (line 22 of every process):

5. 2018-1

```

01  /*
02  Copyright 2018 Rahmat M. Samik-Ibrahim
03  You are free to SHARE (copy and
04  redistribute the material in any medium
05  or format) and to ADAPT (remix,
06  transform, and build upon the material
07  for any purpose, even commercially).
08  This program is distributed in the hope
09  that it will be useful, but WITHOUT ANY
10  WARRANTY; without even the implied
11  warranty of MERCHANTABILITY or FITNESS
12  FOR A PARTICULAR PURPOSE.
13
14  * REVO2 Wed May  2 11:30:19 WIB 2018
15  * START Wed Apr 18 19:50:01 WIB 2018
16  */
17
18  // DO NOT USE THE SAME SEMAPHORE NAME!!!!
19  // Replace "demo" with your own SSO name.
20  #define SEM_COUNT1      "/count-1-demo"
21  #define SEM_COUNT2      "/count-2-demo"
22  #define SEM_MUTEX       "/mutex-demo"
23  #define SEM_SYNC        "/sync-demo"
24
25  #include <fcntl.h>
26  #include <stdio.h>
27  #include <stdlib.h>
28  #include <unistd.h>
29  #include <semaphore.h>
30  #include <sys/mman.h>
31  #include <sys/types.h>
32  #include <sys/wait.h>
33
34  // Shared Memory: R/W with no name.
35  #define PROT      (PROT_READ | PROT_WRITE)
36  #define VISIBLE  (MAP_ANONYMOUS|MAP_SHARED)
37
38  #define LOOP      2
39  #define BUFSIZE  1
40
41  sem_t*   ctr_prod;
42  sem_t*   ctr_cons;
43  sem_t*   mutex;
44  sem_t*   ssync;
45  int*     product;
46
47  // WARNING: NO ERROR CHECK! ////////////////
48  void flushprintf(char* str, int ii) {
49      printf("%s [%d]\n", str, ii);
50      fflush(NULL);
51  }
52
53  void init(void) {
54      product = mmap(NULL, sizeof(int),
55                     PROT, VISIBLE, 0, 0);
56      *product = 0;
57      ctr_prod = sem_open(SEM_COUNT1,
58                          O_CREAT, 0600, BUFSIZE);
59      ctr_cons = sem_open(SEM_COUNT2,
60                          O_CREAT, 0600, 0);
61      mutex    = sem_open(SEM_MUTEX,
62                          O_CREAT, 0600, 1);
63      ssync    = sem_open(SEM_SYNC,
64                          O_CREAT, 0600, 0);
65  }
66
67  void producer (void) {
68      sem_wait(ssync);
69      flushprintf("PRODUCER  PID",getpid());
70      for (int loop=0; loop<LOOP; loop++) {
71          sem_wait(ctr_prod);
72          sem_wait(mutex);
73          flushprintf("PRODUCT  ",
74                      ++(*product));
75          sem_post(mutex);
76          sem_post(ctr_cons);
77      }
78  }
79
80  void consumer (void) {
81      flushprintf("CONSUMER  PID",getpid());
82      sem_post(ssync);
83      for (int loop=0; loop<LOOP; loop++) {
84          sem_wait(ctr_cons);
85          sem_wait(mutex);
86          flushprintf("CONSUME  ", *product);
87          sem_post(mutex);
88          sem_post(ctr_prod);
89      }
90  }
91
92  // WARNING: NO ERROR CHECK! ////////////////
93  void main(void) {
94      flushprintf("STARTING  PID",getpid());
95      init();
96      if (fork()) producer(); // Parent
97      else      consumer(); // Child
98      sem_unlink(SEM_COUNT1);
99      sem_unlink(SEM_COUNT2);
100     sem_unlink(SEM_SYNC);
101     sem_unlink(SEM_MUTEX);
102     flushprintf("STOP HERE PID", getpid());
103  }

```


7. 2018-2

```

001 // FILE: 30-add1sub1.c =====
002 // Copyright (C) 2018 Rahmat M. Samik-Ibrahim.
003 /* You are free to SHARE (copy and redistribute the material
in any medium or format) and to ADAPT (remix, transform, and
build upon the material for any purpose, even commercially). This
program is distributed in the hope that it will be useful, but WITH-
OUT ANY WARRANTY; without even the implied warranty of MER-
CHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. */

005 // REV04 Sun Dec 16 11:15:54 WIB 2018
006 // START Wed Nov 14 20:30:05 WIB 2018
007
008 #include <fcntl.h>
009 #include <stdio.h>
010 #include <stdlib.h>
011 #include <string.h>
012 #include <semaphore.h>
013 #include <unistd.h>
014 #include <sys/mman.h>
015 #include <sys/types.h>
016 #include <sys/stat.h>
017 #include <sys/wait.h>
018
019 #define MYFLAGS      O_CREAT | O_RDWR
020 #define MYPROTECT PROT_READ | PROT_WRITE
021 #define MYVISIBILITY      MAP_SHARED
022 #define SFILE          "demo-file.bin"
023
024 typedef struct {
025     sem_t  sync[3];
026     int    share;
027     int    loop;
028     pid_t  relative;
029 } myshare;
030
031 myshare* mymap;
032
033 void flushprintf(char* tag1, char* tag2){
034     printf("%s[%s] loop%d relative(%d)\n",
035           tag1, tag2, mymap->loop,
036           getpid() + mymap->relative);
037     fflush(NULL);
038 }

040 #define MAIN "30:ADDSUB"
041 #define ADD1 " 31:ADD1"
042 #define SUB1 " 32:SUB1"
043
044 void main(void) {
045     int fd  =open(SFILE,MYFLAGS,S_IRWXU);
046     int ssize=sizeof(myshare);
047     truncate(SFILE, ssize);
048     mymap=mmap(NULL, ssize, MYPROTECT,
049               MYVISIBILITY, fd, 0);
050     mymap->share  = 0;
051     mymap->loop   = 3;
052     mymap->relative = 1000 - getpid();

053     sem_init (&(mymap->sync[0]), 1, 0);
054     sem_init (&(mymap->sync[1]), 1, 0);
055     sem_init (&(mymap->sync[2]), 1, 0);

056     flushprintf(MAIN, "EXEC");
057     if (!fork())
058         execlp("./31-add1", ADD1, NULL);
059     if (!fork())
060         execlp("./32-sub1", SUB1, NULL);
061     do {
062         sleep(1);
063         flushprintf(MAIN, "LOOP");
064     } while (--mymap->loop);

065     sem_wait (&(mymap->sync[0]));
066     sem_wait (&(mymap->sync[0]));
067     flushprintf(MAIN, "WAIT");
068     if      (mymap->share > 1500)
069         flushprintf("SHARE +/-", "2000");
070     else if (mymap->share > 500)
071         flushprintf("SHARE +/-", "1000");
072     else
073         flushprintf("SHARE +/-", "0");
074     wait(NULL);
075     wait(NULL);
076     flushprintf(MAIN, "EXIT");
077     close(fd);
078 }

```

(a) What is the purpose of line 37?
