

## 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 }
```

- (KOLOM) Lingkari tahun angkatan anda berikut ini: (A) 2016 (B) 2015 (C) lainnya.
- (BARIS) Lingkari sesuai angka terakhir (paling kanan) dari NPM anda: 0 1 2 3 4 5 6
- Harap mengisi (KOLOM:BARIS) dengan 1000
- 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	
<pre> 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 &lt;stdio.h&gt; 017 #include &lt;unistd.h&gt; 018 #include &lt;sys/types.h&gt; </pre>	<pre> 019 #include &lt;sys/wait.h&gt; 020 #include &lt;stdlib.h&gt; 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 } </pre>

## Program Output (line 34 of every process):

R e l P I D :

-----

-----

## 4. (6 points) 2017-2

The Program Code	
<pre> 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 &lt;stdio.h&gt; 010 #include &lt;unistd.h&gt; 011 #include &lt;sys/types.h&gt; 012 #include &lt;sys/wait.h&gt; 013 014 #define LOOP 3 015 #define OFFSET 1000 </pre>	<pre> 017 void main(void) { 018     int basePID = getpid() - OFFSET; 019 020     for (int ii=0; ii &lt; LOOP; ii++) { 021         if(!fork()) { 022             printf("PID[%d]-PPID[%d]\n", 023                 getpid() - basePID, 024                 getppid() - basePID); 025             fflush(NULL); 026         } 027     } 028 } </pre>

## 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      wait(NULL);
79  }
80
81  void consumer (void) {
82      flushprintf("CONSUMER  PID",getpid());
83      sem_post(ssync);
84      for (int loop=0; loop<LOOP; loop++) {
85          sem_wait(ctr_cons);
86          sem_wait(mutex);
87          flushprintf("CONSUME  ", *product);
88          sem_post(mutex);
89          sem_post(ctr_prod);
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  }

```

