This set of Tricky and Buggy questions and answers focuses on PThreads handling.

1. What is the output of this program

```
1.  #include<stdio.h>
2.  #include<pthread.h>
3.
4.  void *fun_t(void *arg);
5.  void *fun_t(void *arg)
6.  {
7.      pthread_exit("Bye");
8.  }
9.  int main()
10.  {
11.      pthread_t pt;
12.      void *res_t;
13.      int ret;
14.      ret = pthread_join(pt,&res_t);
15.      printf("%d\n",ret);
16.      return 0;
17.  }
```

a) 0

b) -1

c) 2

d) 3

View Answer

Answer: d
Explanation: The function pthread_join() returns the error number on error.
Output:
[root@localhost sanfoundry]# gcc -o san san.c -lpthread
[root@localhost sanfoundry]# ./san
3
[root@localhost sanfoundry]#

2. What is the output of this program?

```
1.  #include<stdio.h>
2.  #include<pthread.h>
3.
4.  sem_t st;
5.  void *fun_t(void *arg);
6.  void *fun_t(void *arg)
7.  {
8.      pthread_exit("Bye");
9.  }
```

```
10.   int main()
11.   {
12.       pthread_t pt;
13.       void *res_t;
14.       if(pthread_create(&pt,NULL,fun_t,NULL) == -1)
15.           perror("pthread_create");
16.       if(sem_init(&st,1,2) != 0)
17.           perror("sem_init");
18.       if(pthread_join(pt,&res_t) == -1)
19.           perror("pthread_join");
20.       if(sem_destroy(&st) != 0)
21.           perror("sem_destroy");
22.       return 0;
23.   }
```

a) this program will print nothing

b) this program will give an error

c) this program will give segmentation fault

d) none of the mentioned

View Answer

Answer: b
Explanation: The header file semaphore.h is required for the function sem_init.
Output:
[root@localhost sanfoundry]# gcc -o san san.c -lpthread
san.c:4:4: error: unknown type name ‘sem_t’
[root@localhost sanfoundry]#

3. What is the output of this program?

```
1.   #include<stdio.h>
2.   #include<pthread.h>
3.   #include<semaphore.h>
4.
5.   sem_t st;
6.   void *fun_t(void *arg);
7.   void *fun_t(void *arg)
8.   {
9.       pthread_exit("Bye");
10.   }
11.   int main()
12.   {
13.       pthread_t pt;
14.       void *res_t;
15.       if(pthread_create(&pt,NULL,fun_t,NULL) == -1)
```

```
16.            perror("pthread_create");
17.        if(sem_init(st,1,2) != 0)
18.            perror("sem_init");
19.        if(pthread_join(pt,&res_t) == -1)
20.            perror("pthread_join");
21.        if(sem_destroy(&st) != 0)
22.            perror("sem_destroy");
23.        return 0;
24.    }
```

a) this program will print nothing

b) this program will give an error

c) this program will give segmentation fault

d) none of the mentioned

View Answer

Answer: b
Explanation: The first arguement of the sem_init() if of type sem_t*.
Output:
[root@localhost sanfoundry]# gcc -o san san.c -lpthread
2_thread.c: In function 'main':
2_thread.c:17:2: error: incompatible type for argument 1 of 'sem_init'
/usr/include/semaphore.h:37:12: note: expected 'union sem_t *' but argument is of type 'sem_t'
[root@localhost sanfoundry]#

4. What is the output of this program?

```
1.    #include<stdio.h>
2.    #include<pthread.h>
3.    #include<semaphore.h>
4.
5.    void *fun_t(void *arg);
6.    void *fun_t(void *arg)
7.    {
8.        pthread_exit("Bye");
9.    }
10.    int main()
11.    {
12.        pthread_t pt;
13.        sem_t st;
14.        void *res_t;
15.        if(pthread_create(&pt,NULL,fun_t,NULL) == -1)
16.            perror("pthread_create");
17.        if(sem_init(&st,0,0) != 0)
18.            perror("sem_init");
19.        if(sem_wait(&st) != 0)
```

```
            perror("sem_wait");
        printf("Sanoundry\n");
        if(pthread_join(pt,&res_t) == -1)
            perror("pthread_join");
        if(sem_destroy(&st) != 0)
            perror("sem_destroy");
        return 0;
    }
```

a) this program will print the string “Sanfoundry”

b) this program will give segmentation fault

c) this process will remain block

d) none of the mentioned

View Answer

Answer: c

Explanation: In this program, initial value of semaphore is 0. The sem_wait() function call blocks the process until the value of semaphore becomes 1.

Output:

```
[root@localhost sanfoundry]# gcc -o san san.c -lpthread
[root@localhost sanfoundry]# ./san
^Z
[3]+ Stopped ./san
[root@localhost sanfoundry]#
```

5. What is the output of this program?

```
#include<stdio.h>
#include<pthread.h>
#include<semaphore.h>

void *fun_t(void *arg);
void *fun_t(void *arg)
{
    sem_post(&st);
    pthread_exit("Bye");
 }
 int main()
 {
     pthread_t pt;
     sem_t st;
     void *res_t;
     if(pthread_create(&pt,NULL,fun_t,NULL) == -1)
         perror("pthread_create");
     if(sem_init(&st,0,0) != 0)
         perror("sem_init");
```

```
20.        if(sem_wait(&st) != 0)
21.            perror("sem_wait");
22.        printf("Sanoundry\n");
23.        if(pthread_join(pt,&res_t) == -1)
24.            perror("pthread_join");
25.        if(sem_destroy(&st) != 0)
26.            perror("sem_destroy");
27.        return 0;
28.    }
```

a) this program will print the string “Sanfoundry”

b) this program will give an error

c) this program will print the string “Sanfoundry” & gives an error

d) none of the mentioned

View Answer

Answer: b
Explanation: The semaphore object st is locally declared.
Output:
[root@localhost sanfoundry]# gcc -o san san.c -lpthread
san.c: In function ‘fun_t’:
san.c:8:12: error: ‘st’ undeclared (first use in this function)
san.c:8:12: note: each undeclared identifier is reported only once for each function it appears in
[root@localhost sanfoundry]#

6. Which one of the following string will print first by this program?

```
1.    #include<stdio.h>
2.    #include<pthread.h>
3.    #include<semaphore.h>
4.
5.    sem_t st;
6.    void *fun_t(void *arg);
7.    void *fun_t(void *arg)
8.    {
9.        printf("Linux\n");
10.        sem_post(&st);
11.        pthread_exit("Bye");
12.    }
13.    int main()
14.    {
15.        pthread_t pt;
16.        void *res_t;
17.        if(pthread_create(&pt,NULL,fun_t,NULL) == -1)
18.            perror("pthread_create");
19.        if(sem_init(&st,0,0) != 0)
```

```
20.            perror("sem_init");
21.        if(sem_wait(&st) != 0)
22.            perror("sem_wait");
23.        printf("Sanoundry\n");
24.        if(pthread_join(pt,&res_t) == -1)
25.            perror("pthread_join");
26.        if(sem_destroy(&st) != 0)
27.            perror("sem_destroy");
28.        return 0;
29.    }
```

a) Linux
b) Sanfoundry
c) Can not be predicted
d) None of the mentioned

View Answer

Answer: a
Explanation: The string “Linux” will print first because of semaphores.
Ouptut:

```
[root@localhost sanfoundry]# gcc -o san san.c -lpthread
[root@localhost sanfoundry]# ./san
Linux
Sanoundry
[root@localhost sanfoundry]#
```


7. What is the output of this program?

```
1.    #include<stdio.h>
2.    #include<pthread.h>
3.    #include<semaphore.h>
4.
5.    sem_t st;
6.    void *fun_t(void *arg);
7.    void *fun_t(void *arg)
8.    {
9.        printf("Linux\n");
10.        pthread_exit("Bye");
11.        sem_post(&st);
12.    }
13.    int main()
14.    {
15.        pthread_t pt;
16.        void *res_t;
```

```
17.        if(pthread_create(&pt,NULL,fun_t,NULL) == -1)
18.            perror("pthread_create");
19.        if(sem_init(&st,0,0) != 0)
20.            perror("sem_init");
21.        if(sem_wait(&st) != 0)
22.            perror("sem_wait");
23.        printf("Sanoundry\n");
24.        if(pthread_join(pt,&res_t) == -1)
25.            perror("pthread_join");
26.        if(sem_destroy(&st) != 0)
27.            perror("sem_destroy");
28.        return 0;
29.    }
```

a) this program will print the only string “Linux”

b) this program will print the only string “Sanfoundry”

c) this program will print both the strings “Linux” and “Sanfoundry”

d) none of the mentioned

View Answer

Answer: a
Explanation: The value of semaphore will never become 1 in this program.
Output:

```
[root@localhost sanfoundry]# gcc -o san san.c -lpthread
[root@localhost sanfoundry]# ./san
Linux
^Z
[4]+ Stopped ./san
[root@localhost sanfoundry]#
```

8. What is the output of this program?

```
1.    #include<stdio.h>
2.    #include<pthread.h>
3.    #include<semaphore.h>
4.
5.    sem_t st;
6.    void *fun_t(void *arg);
7.    void *fun_t(void *arg)
8.    {
9.        printf("Linux\n");
10.        pthread_exit("Bye");
11.    }
12.    int main()
13.    {
14.        pthread_t pt;
```

```
15.        void *res_t;
16.        if(pthread_create(&pt,NULL,fun_t,NULL) == -1)
17.            perror("pthread_create");
18.        if(sem_init(&st,0,2) != 0)
19.            perror("sem_init");
20.        if(sem_wait(&st) != 0)
21.            perror("sem_wait");
22.        printf("Sanoundry\n");
23.        if(sem_wait(&st) != 0)
24.            perror("sem_wait");
25.        if(pthread_join(pt,&res_t) == -1)
26.            perror("pthread_join");
27.        if(sem_destroy(&st) != 0)
28.            perror("sem_destroy");
29.        return 0;
30.    }
```

a) this program will print the only string “Linux”

b) this program will print the only string “Sanfoundry”

c) this program will print both the strings “Linux” and “Sanfoundry”

d) none of the mentioned

View Answer

Answer: c
Explanation: The initial value of semaphore is 2. Hence sem_wait() will only decrement the value of semaphore and the process will not block.
Ouptut:

```
[root@localhost sanfoundry]# gcc -o san san.c -lpthread
[root@localhost sanfoundry]# ./san
Linux
Sanoundry
[root@localhost sanfoundry]#
```

9. In this program the semaphore

```
1.    #include<stdio.h>
2.    #include<pthread.h>
3.    #include<semaphore.h>
4.
5.    sem_t st;
6.    void *fun_t(void *arg);
7.    void *fun_t(void *arg)
8.    {
9.        pthread_exit("Bye");
10.    }
11.    int main()
```

```
{
    pthread_t pt;
    void *res_t;
    if(pthread_create(&pt,NULL,fun_t,NULL) == -1)
        perror("pthread_create");
    if(sem_init(&st,1,2) != 0)
        perror("sem_init");
    if(pthread_join(pt,&res_t) == -1)
        perror("pthread_join");
    if(sem_destroy(&st) != 0)
        perror("sem_destroy");
    return 0;
}
```

a) can be used only for this process
b) can be used for any other process also
c) can be used
d) none of the mentioned

View Answer

Answer: b
Explanation: The value of second argument of sem_init() is 1.
Output:
[root@localhost sanfoundry]# gcc -o san san.c -lpthread
[root@localhost sanfoundry]# ./san
[root@localhost sanfoundry]#

10. Which one of the following string will print by this program?

```
#include<stdio.h>
#include<pthread.h>
int main()
{
    printf("Sanfoundry\n");
    pthread_exit("Bye");
    printf("Linux");
    return 0;
}
```

a) Linux
b) Sanfoundry
c) Bye
d) None of the mentioned

View Answer

Answer: b
Explanation: The main thread exits before printing the string “Linux”;
Output:

```
[root@localhost sanfoundry]# gcc -o san san.c -lpthread
[root@localhost sanfoundry]# ./san
Sanfoundry
[root@localhost sanfoundry]#
```