This set of Linux Debugging questions and answers focuses on Posix Threads.

1. Which one of the following string will print first by 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.      printf("Sanfoundry\n");
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) != 0)
15.          perror("pthread_create");
16.      printf("Linux\n");
17.      if(pthread_join(pt,&res_t) != 0)
18.          perror("pthread_join");
19.      return 0;
20.  }
```

a) Linux
b) Sanfoundry
c) it can not be predicted
d) none of the mentioned

View Answer

Answer: b
Explanation: It depends upon the scheduler.
Output:

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

2. 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.      int ret;
```

```
8.        ret = pthread_exit("Bye");
9.        printf("%d\n",ret);
10.   }
11.   int main()
12.   {
13.       pthread_t pt;
14.       void *res_t;
15.       if(pthread_create(&pt,NULL,fun_t,NULL) != 0)
16.           perror("pthread_create");
17.       if(pthread_join(pt,&res_t) != 0)
18.           perror("pthread_join");
19.       return 0;
20.   }
```

a) 0
b) 1
c) -1
d) none of the mentioned
View Answer
Answer: d
Explanation: The function pthread_exit() does not return any value. Hence this program will give an error.
Output:
[root@localhost sanfoundry]# gcc -o san san.c -lpthread
san.c: In function 'fun_t':
san.c:8:6: error: void value not ignored as it ought to be
[root@localhost sanfoundry]#

3. 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.       printf("Sanfoundry\n");
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) != 0)
15.           perror("pthread_create");
16.       return 0;
17.   }
```

a) this program will print the string “Sanfoundry”
b) this program will print nothing
c) segmentation fault
d) run time error
View Answer
Answer: b
Explanation: The pthread_join() function waits for the thread to terminate.
Output:
[root@localhost sanfoundry]# gcc -o san san.c -lpthread
[root@localhost sanfoundry]# ./san
[root@localhost sanfoundry]#

4. 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.      printf("%d\n",a);
8.      pthread_exit("Bye");
9.  }
10.  int main()
11.  {
12.      int a;
13.      pthread_t pt;
14.      void *res_t;
15.      a = 10;
16.      if(pthread_create(&pt,NULL,fun_t,NULL) != 0)
17.          perror("pthread_create");
18.      if(pthread_join(pt,&res_t) != 0)
19.          perror("pthread_join");
20.      return 0;
21.  }
```

a) 10
b) 0
c) -1
d) none of the mentioned
View Answer
Answer: d
Explanation: Each thread has its own stack so local variables are not shared among thread. Hence this program will give an error.
Output:
[root@localhost sanfoundry]# gcc -o san san.c -lpthread
san.c: In function ‘fun_t’:

san.c:7:16: error: ‘a’ undeclared (first use in this function)
san.c:7:16: note: each undeclared identifier is reported only once for each function it appears in
[root@localhost sanfoundry]#

5. What is the output of this program?

```
1.  #include<stdio.h>
2.  #include<pthread.h>
3.
4.  int a;
5.  void *fun_t(void *arg);
6.  void *fun_t(void *arg)
7.  {
8.      printf("%d\n",a);
9.      pthread_exit("Bye");
10.  }
11.  int main()
12.  {
13.      pthread_t pt;
14.      void *res_t;
15.      a = 10;
16.      if(pthread_create(&pt,NULL,fun_t,NULL) != 0)
17.          perror("pthread_create");
18.      if(pthread_join(pt,&res_t) != 0)
19.          perror("pthread_join");
20.      return 0;
21.  }
```

a) 10
b) 0
c) -1
d) none of the mentioned
View Answer
Answer: a
Explanation: Thread of the same process shares the global variables.
Output:
[root@localhost sanfoundry]# gcc -o san san.c -lpthread
[root@localhost sanfoundry]# ./san
10
[root@localhost sanfoundry]#

6. What is the output of this program?

```
1.  #include<stdio.h>
2.  #include<pthread.h>
3.
4.  int a;
```

```
5.   void *fun_t(void *arg);
6.   void *fun_t(void *arg)
7.   {
8.       a = 20;
9.       pthread_exit("Bye");
10.   }
11.   int main()
12.   {
13.       pthread_t pt;
14.       void *res_t;
15.       a = 10;
16.       if(pthread_create(&pt,NULL,fun_t,NULL) != 0)
17.           perror("pthread_create");
18.       if(pthread_join(pt,&res_t) != 0)
19.           perror("pthread_join");
20.       printf("%d\n",a);
21.       return 0;
22.   }
```

a) 10
b) 20
c) segmentation fault
d) none of the mentioned
View Answer
Answer: b
Explanation: In this program the value of variable “a” is changed by the thread “fun_t”.
Output:

```
[root@localhost sanfoundry]# gcc -o san san.c -lpthread
[root@localhost sanfoundry]# ./san
20
[root@localhost sanfoundry]#
```


7. Which one of the following statement is not true about 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.       printf("%d\n",getpid());
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) != 0)
15.           perror("pthread_create");
16.       if(pthread_join(pt,&res_t) != 0)
17.           perror("pthread_join");
18.       printf("%d\n",getpid());
19.       return 0;
20.   }
```

a) both printf statements will print the same value
b) both printf statements will print the different values
c) this program will print nothing
d) none of the mentioned
View Answer
Answer: a
Explanation: All the threads of the same process have same PID.
Output:
[root@localhost sanfoundry]# gcc -o san san.c -lpthread
[root@localhost sanfoundry]# ./san
12981
12981
[root@localhost sanfoundry]#

8. What is the output of this program?

```
1.   #include<stdio.h>
2.   #include<pthread.h>
3.   #include<fcntl.h>
4.
5.   int fd;
6.   void *fun_t(void *arg);
7.   void *fun_t(void *arg)
8.   {
9.       char buff[10];
10.       int count;
11.       count = read(fd,buff,10);
12.       printf("%d\n",count);
13.       pthread_exit("Bye");
14.   }
15.   int main()
16.   {
17.       pthread_t pt;
18.       void *res_t;
19.       fd = open("san.c",O_RDONLY);
20.       if(pthread_create(&pt,NULL,fun_t,NULL) != 0)
21.           perror("pthread_create");
```

```
22.         if(pthread_join(pt,&res_t) != 0)
23.             perror("pthread_join");
24.         return 0;
25.     }
```

a) 10
b) 0
c) -1
d) segmentation fault
View Answer

Answer: a
Explanation: Open file descritpors can be shares between threads of the same process
Output:

```
[root@localhost sanfoundry]# gcc -o san san.c -lpthread
[root@localhost sanfoundry]# ./san
10
[root@localhost sanfoundry]#
```

9. What is the output of this program?

```
1.    #include<stdio.h>
2.    #include<pthread.h>
3.    #include<fcntl.h>
4.
5.    void *fun_t(void *arg);
6.    void *fun_t(void *arg)
7.    {
8.        pthread_exit("Bye");
9.        printf("Sanfoundry\n");
10.    }
11.    int main()
12.    {
13.        pthread_t pt;
14.        void *res_t;
15.        if(pthread_create(&pt,NULL,fun_t,NULL) != 0)
16.            perror("pthread_create");
17.        if(pthread_join(pt,&res_t) != 0)
18.            perror("pthread_join");
19.        printf("%s\n",res_t);
20.        return 0;
21.    }
```

a) Sanfoundry
b) Bye
c) segementation fault
d) run time error
View Answer

Answer: b
Explanation: None.
Output:
[root@localhost sanfoundry]# gcc -o san san.c -lpthread
[root@localhost sanfoundry]# ./san
Bye
[root@localhost sanfoundry]#

10. 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.        sleep(1);
8.    }
9.    int main()
10.    {
11.        pthread_t pt;
12.        void *res_t;
13.        if(pthread_create(&pt,NULL,fun_t,NULL) != 0)
14.            perror("pthread_create");
15.        if(pthread_join(pt,&res_t) != 0)
16.            perror("pthread_join");
17.        printf("%s\n",res_t);
18.        return 0;
19.    }
```

a) this process will pause
b) segmentation fault
c) run time error
d) none of the mentioned
View Answer
Answer: b
Explanation: This program is trying to print the return value of the thread, but pthread_exit() function is not present in the thread.
Output:
[root@localhost sanfoundry]# gcc -o san san.c -lpthread
[root@localhost sanfoundry]# ./san
Segmentation fault (core dumped)
[root@localhost sanfoundry]#