P1.c

1 #include<stdio.h>

2 main()

3 {

4 printf("hi...\n");

5 vfork();

6 printf("hello...\n");

7 exit(0); //if not written exit()\_exit undefined behaviour

8 }

P2.c

1 #include<stdio.h>

2 main()

3 {

4 printf("hi...\n");

5 vfork();

6 printf("hello...\n");

7 return; //writing return in a function where vfork() called results undefined

8 exit(0); //behaviour

9 }

P3.c

1 #include<stdio.h>

2 main()

3 {

4

5 if(vfork()==0)

6 {

7 printf("in child:%d\n",getpid());

8 /\*sleep(10);

9 printf("child terminated...\n");

10 exit(0);\*/

11 while(1);

12 }

13 else

14 {

15 printf("in parent :%d\n",getpid());

16 /\*sleep(1);

17 printf("parent terminated...\n");

18 exit(0);\*/

19 while(1);

20 }

21 }

P4.c

1 #include<stdio.h>

2 int i=1;

3 main()

4 {

5 printf("hi..\n");

6 vfork();

7 i++;

8 printf("i=%d\n",i);

9 exit(0);

10 }

11 //in this example after vfork first child process executed and modifies the i

12 //value it is effected to parent process also (because there is no cow concept)

13 //after that parent also incremented i value.

14 //if child modifies parent will effect ,if parent modifies child will effect