```
#include <fcntl.h>
#include <stdio.h>
main()
{  int fd;  /* file descriptor */    int pid; /* process id */
   char buf[10]; /* buffer to hold file data */
   if ( (fd = open(“a.txt”, O_RDONLY) ) < 0 )
     {       printf(“file open error₩n”); exit(1);
     }
   read(fd, buf, 10);
   printpos(“Before fork”, fd);
   if ( (pid = fork() ) < 0 )
      {         printf(“fork error₩n”);  exit(1);   }
   else if ( pid == 0 )   /* child */
       {   printpos(“Child before read”, fd);
            read(fd, buf, 10);
            printpos(“Child after read”, fd);
        }
   else   /* parent */
        {    wait( (int *)0 );
              printpos(“Parent after wait”, fd);
         }
  }
```

```
printpos(string, filedes)
char *string;
int filedes;
{
    long pos, lseek();
    if ( (pos = lseek(filedes, 0L, 1)) < 0L)
          printf(“error: lseek failed₩n”);
   printf(“%s: %ld₩n”, string, pos);
}
```

<결과>

```
Before fork : 10
Child before read : 10
child after read : 20
Parent after wait : 20
```

좀비프로세스 발생함

```
#include <stdlib.h>
#include <stdio.h>
main()
{
   int child;
   if((child=fork())==0)
   {
      printf(“child PID %d₩n”, getpid());
      pause(); ← 어떠한 신호가 오기전까지는 중단 (살아있음)
   }
   /* 부모 */
  printf(“child PID %d₩n”, child);
  exit(child); ← 부모 죽음
}
```

해결 : kill PID