Date: 16-11-21:

fcntl()

→ How to duplicate file descriptor?

① dup()
② dup2()
} ✓

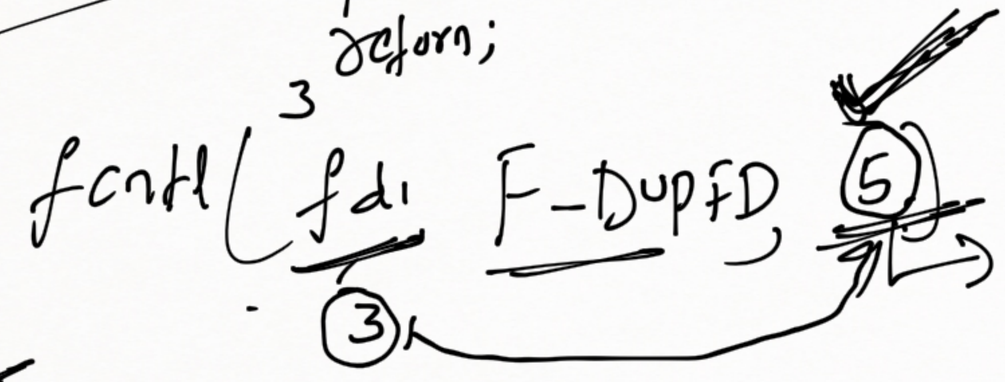
→ We can use fcntl() also for duplicating the file descriptor

```
main()
{
int fd1, fd2;
fd1 = open("Temp", O_WRONLY | O_CREAT | O_TRUNC, 0644)
if(fd1<0)
{perror("open");
exit;
}
fd2 = fcntl(fd1 F_DUPFD 5);
```

fd1 = 3

fd2 = 5

command → F_DUPFD



→ If 5 is free than fd1 is duplicated with 5, suppose 5 is busy then it will duplicate with next number is 6

→ dup3 (fd1, 5);

dup() ✓
dup2() ✓
fcntl() ✓

→ fcntl (fd1, F-DUPFD, 5)

↳ command is useful for duplicating the file descriptor

→ Using fcntl() possible to find size of the pipe without blocking the pipe?

1)

```
main()
{
  int fd[2],count=0;
  char ch='a';
  pipe2(fd, O_NONBLOCK);
  while(write(fd[1],&ch,1)>0)
    count++;
  printf("size of the pipe: %d\n", count);
}
```

→ this flag will nonblock the pipe on both sides (read end/write end)

→ using pipe2 with O_NONBLOCK flag both the ends are nonblock (read end and write end)

→ specific end nonblock not possible with pipe2()

→ Using fcntl() specific end non block is possible

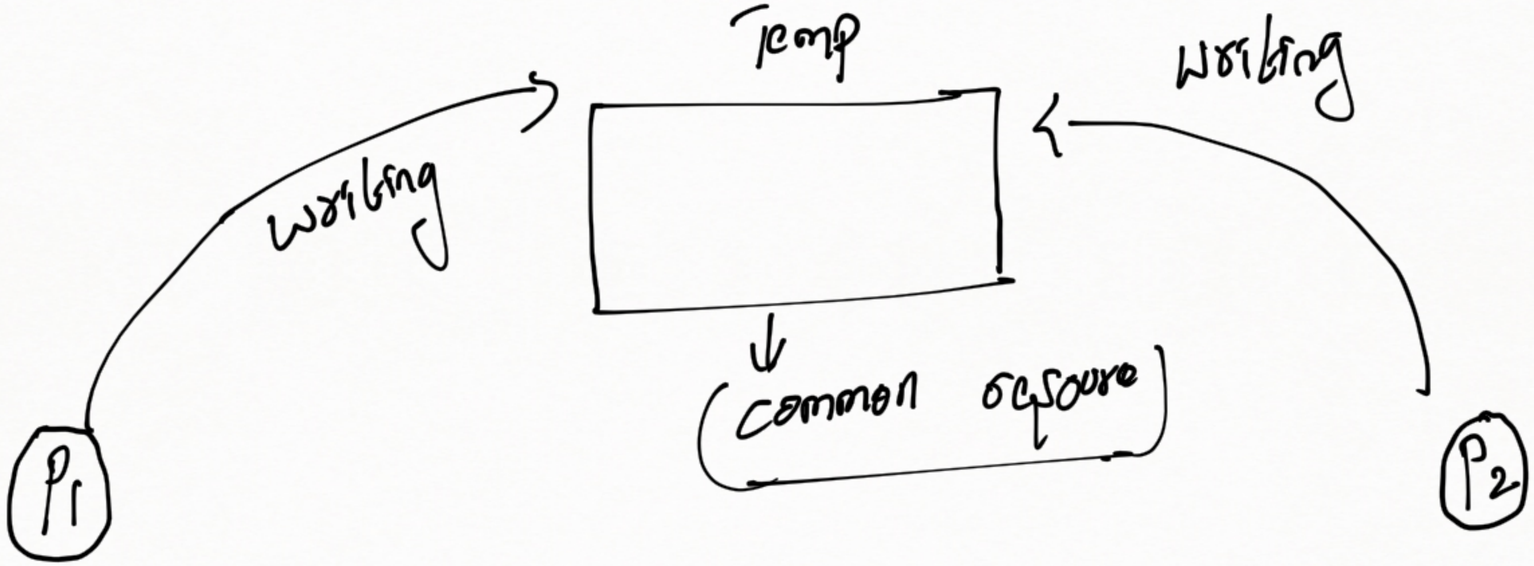
```
main()
{ int fd[2], count=0;
  char ch='a';
  pipe(fd);
fcntl(fd[1], F_SETFL, O_NONBLOCK)
  while( write(fd[1], &ch, 1) >0)
      count++;
  printf("size of the pipe: %d in", count);
}
```

↳ this command will nonblock the pipe on specific end

↳ write end of the pipe is nonblocked

Advisory locking (or) Record locking

→ fcntl() also useful for record locking.



→ when more than one process trying to access a

common resource (file) then there must be a proper synchronization otherwise chance of data corruption.

⇒ By using advisory locking when more then one process trying to access a common resource we can provide proper synchronization struct flock (d)

fcntl (fd, F_SETLK (or) F_SETLKW / F_GETLK, &V)