```
int create(const char* path_name, int permissions);

char* path_name = "/home/yogeshwar/src/hello.c;

int fd = create(path_name, S_IRUSR | S_IWUSR | S_IRGRP | S_IROTH);
if(fd == -1){
    // error
}

#--------
app2.c
which wants to open /home/yogeshwar/src/hello.c

int fd = open(path_name, O_RDONLY);
if(fd == -1){
    // open error
}

#--------------------------------------

int fd;
const char* path_name = "/home/yogeshwar/src/hello.c";

fd = open(path_name, O_RDONLY);
if(fd == -1){    /* hello.c is not present*/
    fd = creat(path_name, S_IRUSR | S_IWUSR | S_IRGRP | S_IROTH);
    if(fd == -1){
        if(errno == EEXISTS){
            // SURPRISE!!!
        }
    }
}

PROCESS 1                  PROCESS 2
open() call
open() failed
conclusion:
base name does not
exist in dir name
PREEMPT
                           create hello.c in /home/yogeshwar/src
                           some other things
                           PREEMPT
```

SCHEDULE

creat()
in order to create hello.c in
/home/yogeshwar/src

creat() will fail with EEXISTS

PROGRAMMER PROPOSES -> PREEMPTIVE MULTITASKING DISPOSES.

```
#-----------------------------------------------------------
```

try to write data at the end of the file.

```
fd = open(path_name, O_WRONLY);
if(fd == -1){
   //error
}

lseek(int fd, int whence, off_t bytes);
lseek(fd, SEEK_SET/SEEK_CUR/SEEK_END, positive/negative);
-------------------------------
lseek(fd, SEEK_END, 0)
write(fd, buffer, BUFFER_SIZE);
```

```
PROCESS 1                                      PROCESS 2

lseek(fd, SEEK_END, 0) -> File table
-> next read write offset -> 22K

PREEMPT
                                               abc.txt
                                                lseek(fd, SEEK_END, 0)
                                                write(fd, buffer, 4096)
                                                PREEMPT
                                                 (26K)
write()
#------------------------------------------------------------------------------
```

Scheduling Atomicity

- Behavioural Atomicity.

P K

system_call_k.

Shared Resource (R)

$R \; : \; R_{final}$

State stati