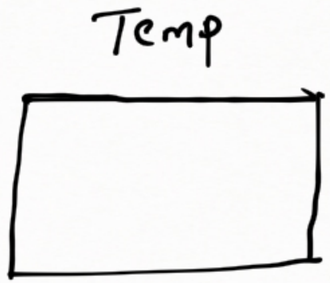
Date: 18-11-21

P1



P2

fcntl (fd, F-SETLKW, &v):

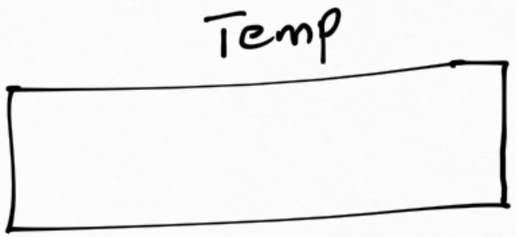
↳ apply lock on file (fd)

when no other process hold lock, But suppose if any other process all ready hold lock then wait until release.

P1

=

=



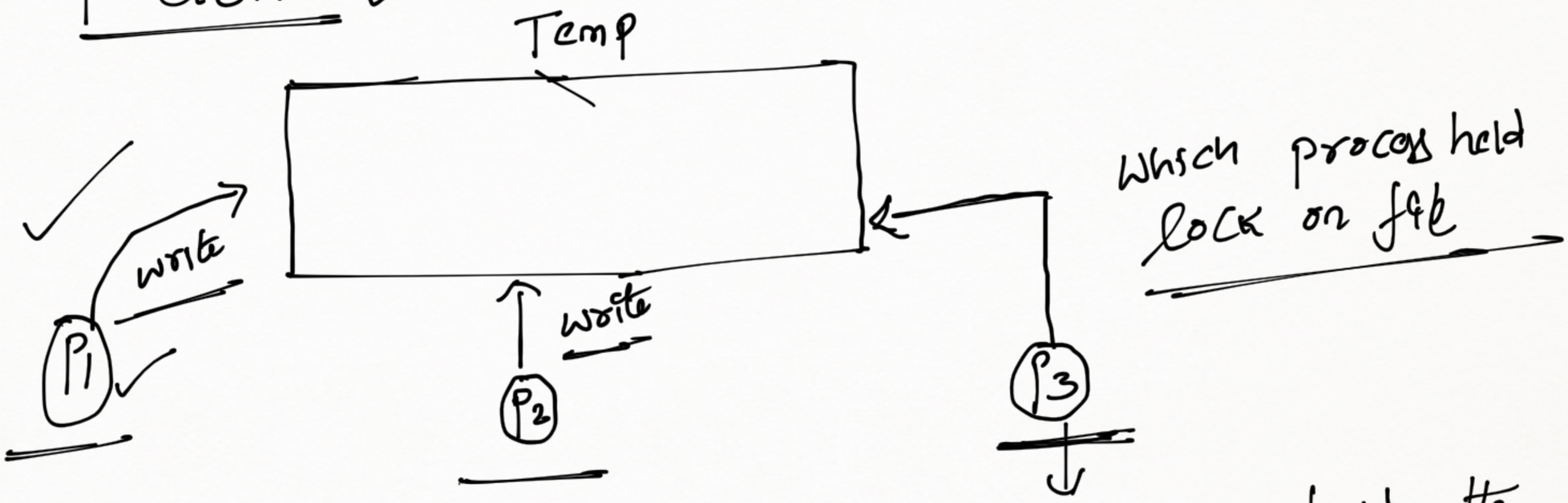
P2

rd = fcntl(fd, F_SETLK, &v);

↳ Apply lock on file (fd)

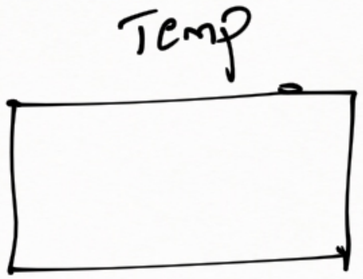
when no other process hold lock, But if a process all ready hold a lock than don't wait fcntl() call fails if returns −1

F-GETLK ✓


Temp
write
write
P1
P2
P3
Which process hold lock on ffb
If no process hold the lock on ffb then apply the lock but already if a process hold the lock then don't wait
P3 want to know that who is blocking this process

P3

```
struct flock v, v1;
v.l_type = F_WRLCK;
v.l_whence = 0;
v.l_start = 0;
v.l_len = 0;
printf("Before fcntl ---\n");
rct = fcntl(fd, F_SETLK, &v);
```




```
if (rct == -1)
{
fcntl(fd, F_GETLK, &v1)
printf("%d", v1.l_pid)
}
else
{
// write data into file.
}
```

process id

→ What is critical section of code?

A) Critical section of code is part of the program where a common resource is accessed, critical section code execution should be atomic (when one process critical section of code is executed then other process should not execute).

→ fcntl() function we can use for synchronization only if the resource is file

O_EXCL flag

↳ (Exclusive creation)

```
fd = open("Temp", O_CREAT | O_WRONLY | O_TRUNC, 0644);
```

```
fd = open("Temp", O_CREAT | O_EXCL | O_WRONLY | O_TRUNC, 0644);
```

↓

When this flag is used if temp file not there if will create but temp file all ready there open() call fails

Message Queues

→ To use message queues for process communication the following functions are required

(i) msgget ⇒ it will create message queue (or) opens the existing queue.

(ii) msgsnd() ⇒ this function useful to send the message into queue.

(iii) msgrcv() ⇒ this function useful to receive the messages from the queue.

(iv) msgctl() ⇒ useful for message Queue control operations

* → When a message Queue is created each Queue associated with key value.