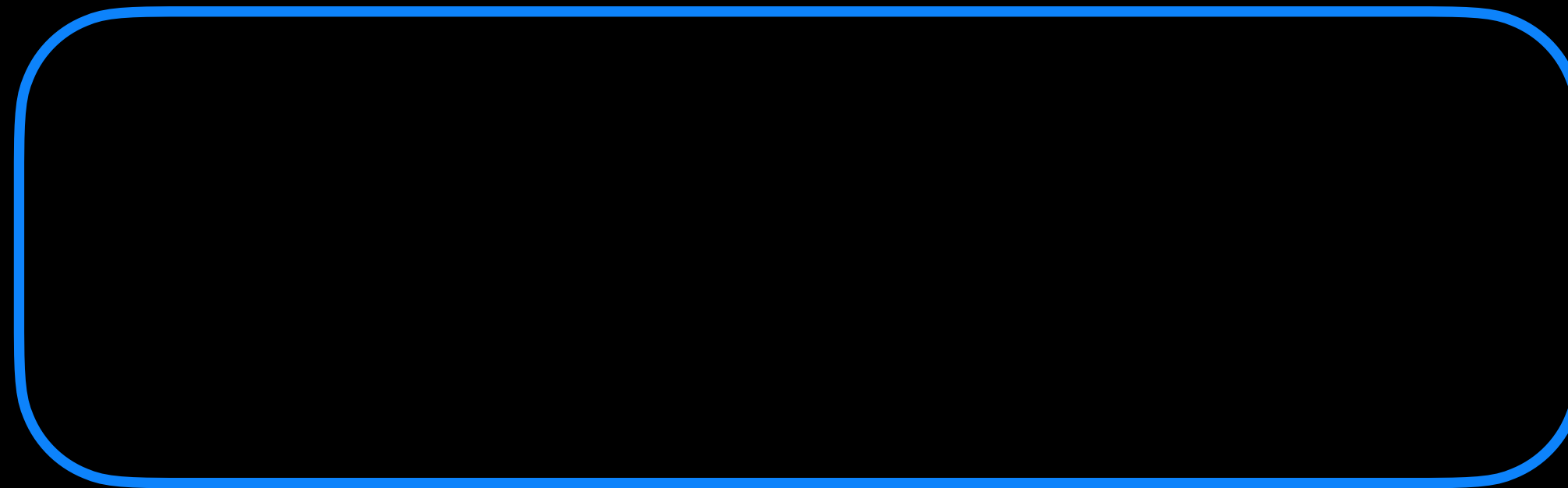


File I/O Buffering

kernel space

user space

process





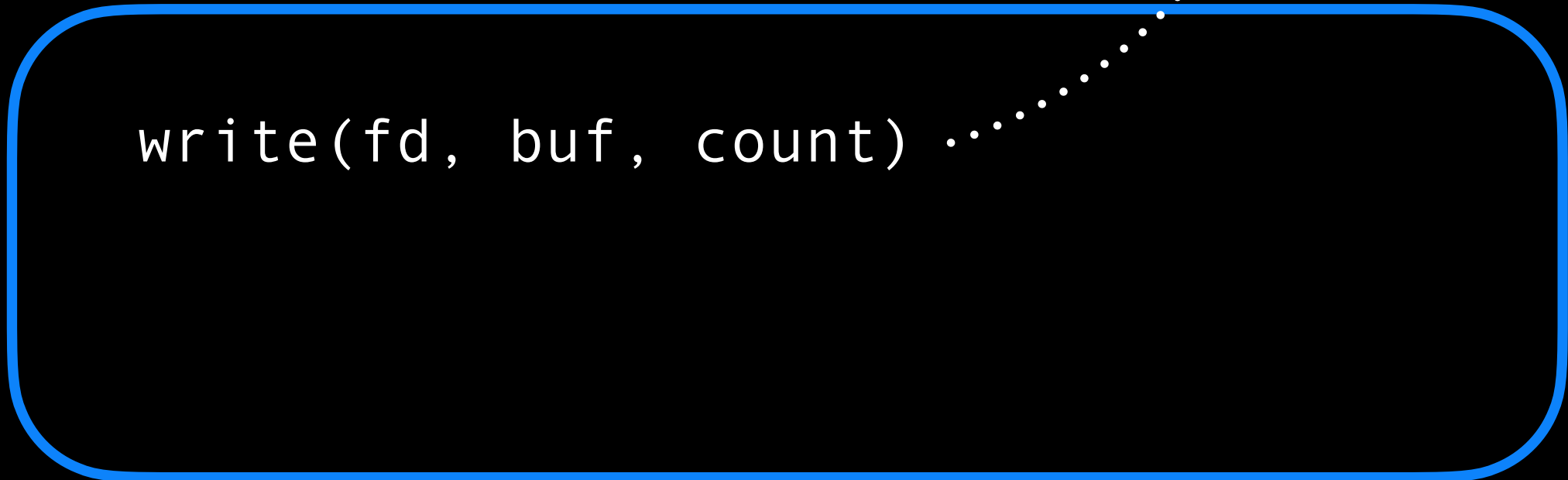
data blocks (on disk)



kernel space

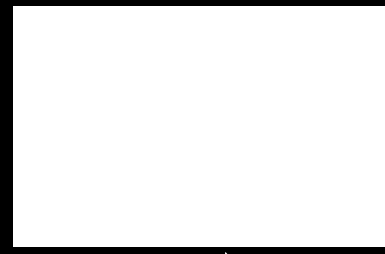
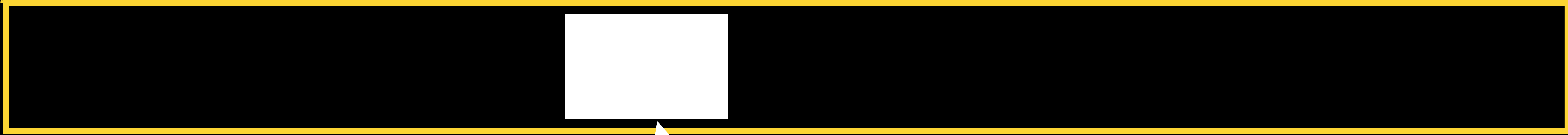
user space

process





data blocks (on disk)



kernel space

user space

process

```
write(fd, buf, count)
```



data blocks (on disk)



buffer cache (in memory)



kernel space

user space

process

```
write(fd, buf, count)
```



data blocks (on disk)



buffer cache (in memory)



kernel space

user space

process

```
write(fd, buf, count)
```



kernel space

user space

process

```
write(fd, buf, count)
...
write(fd, buf, count)
```



data blocks (on disk)



buffer cache (in memory)

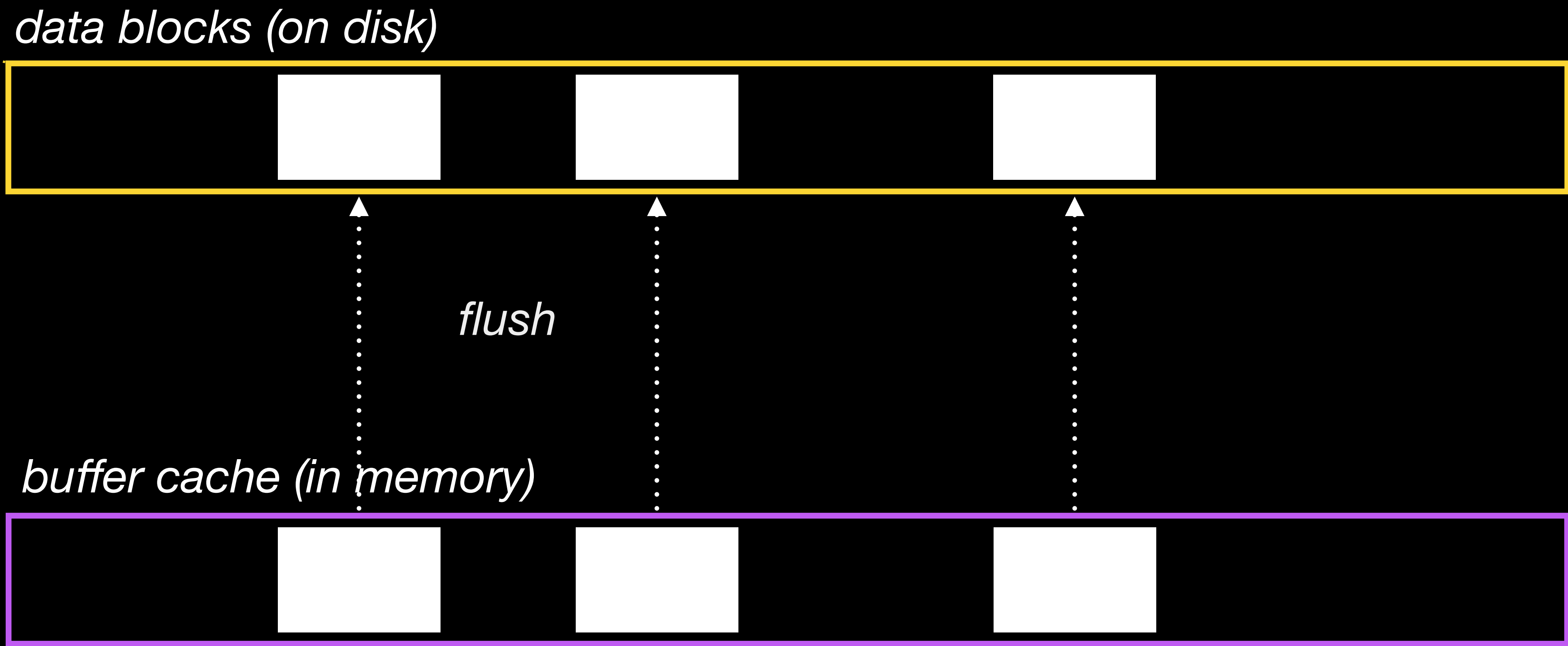


kernel space

user space

process

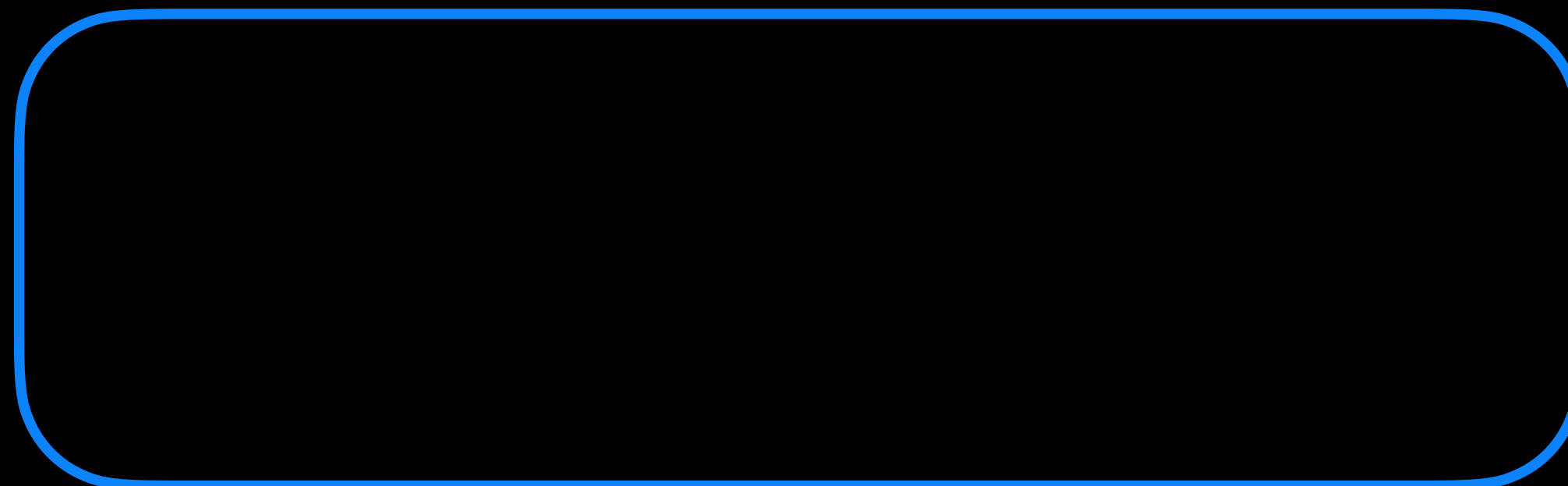
```
write(fd, buf, count)
write(fd, buf, count)
write(fd, buf, count)
...
```

kernel space

user space

process





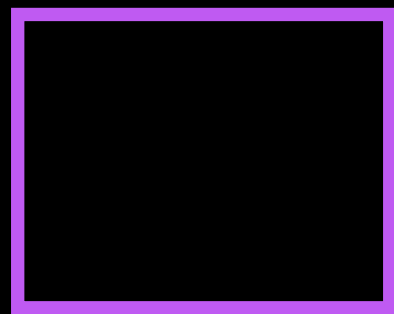
data blocks (on disk)



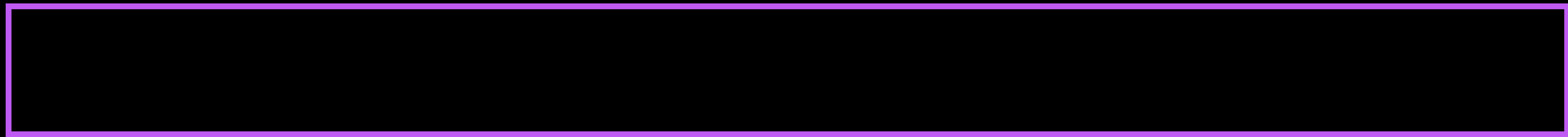
i-node table (on disk)



*i-node
(in-memory)*



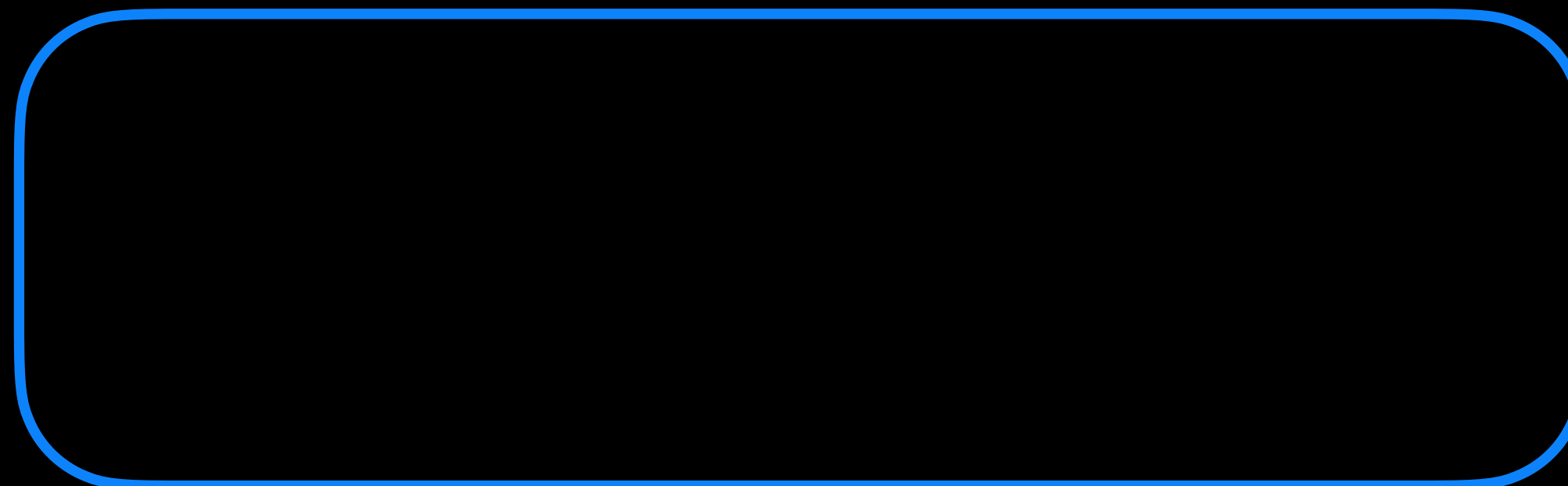
buffer cache (in memory)

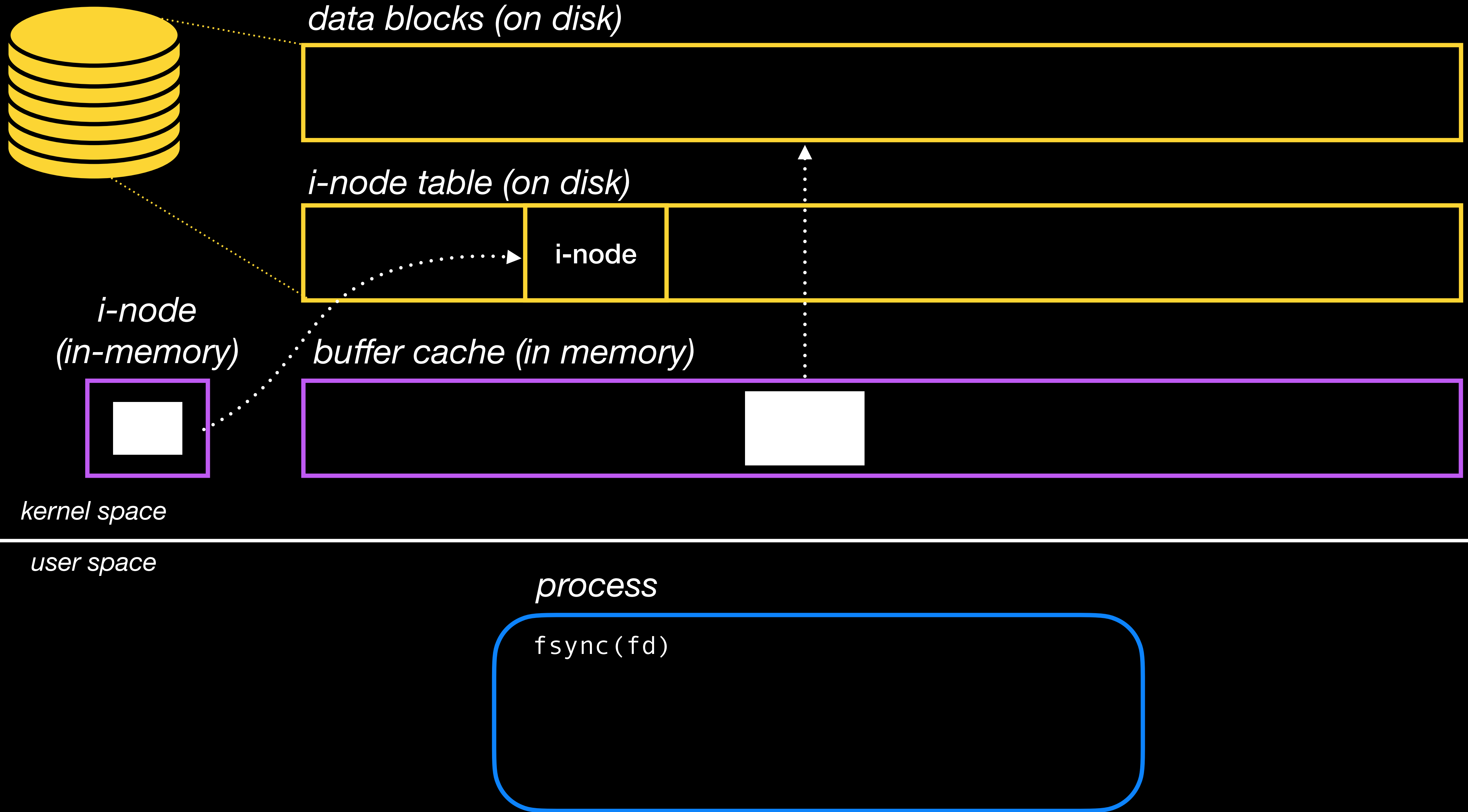


kernel space

user space

process







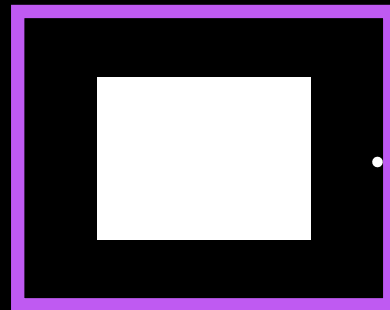
data blocks (on disk)



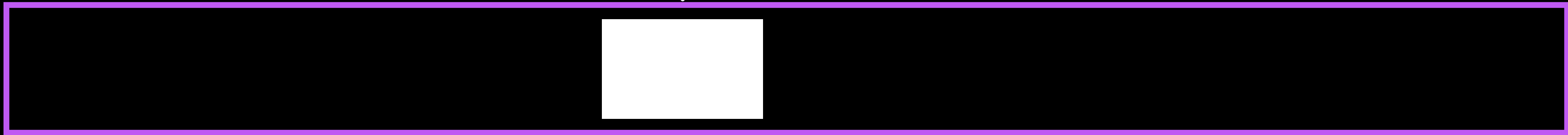
i-node table (on disk)



*i-node
(in-memory)*



buffer cache (in memory)



kernel space

user space

process

`fsync(fd)`

`syncfs(fd)`

All files on the
same filesystem as fd

`sync()`

All files on
all filesystems

`open(path, O_SYNC)`

All writes implicitly
call fsync()



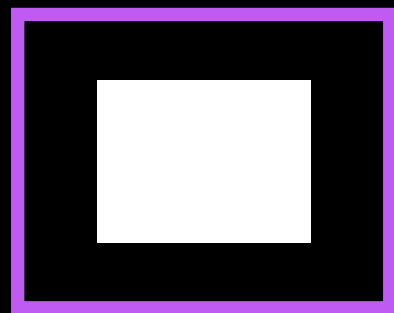
data blocks (on disk)



i-node table (on disk)



*i-node
(in-memory)*



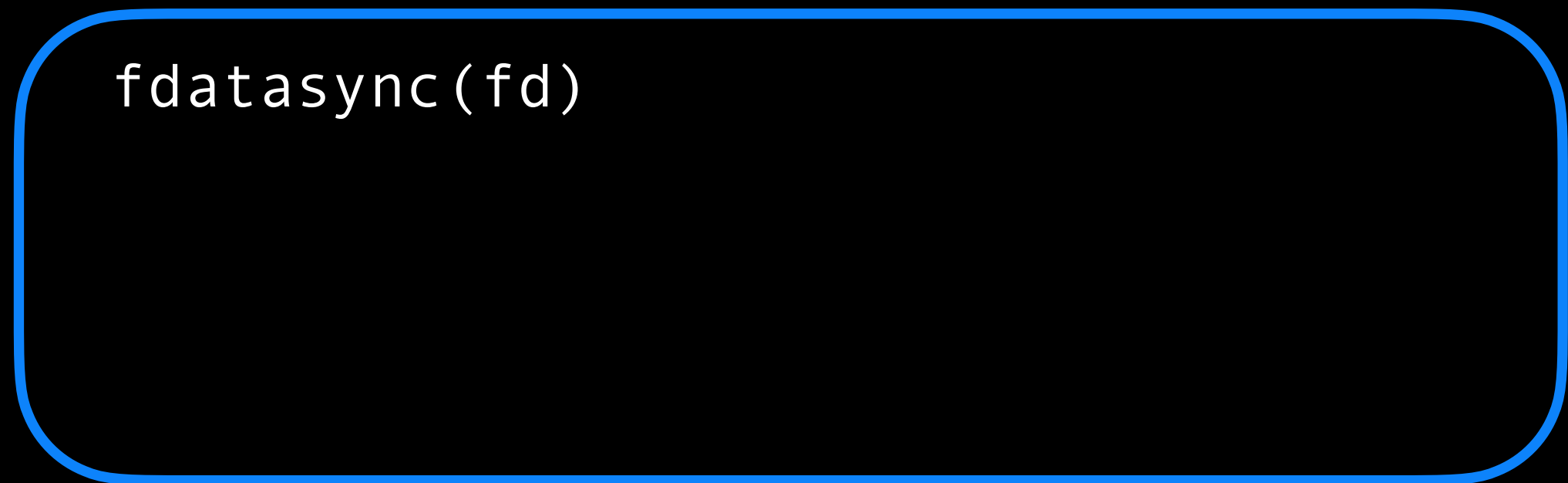
buffer cache (in memory)



kernel space

user space

process





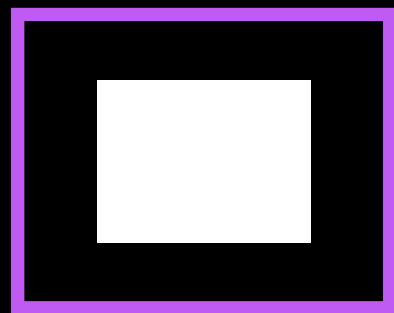
data blocks (on disk)



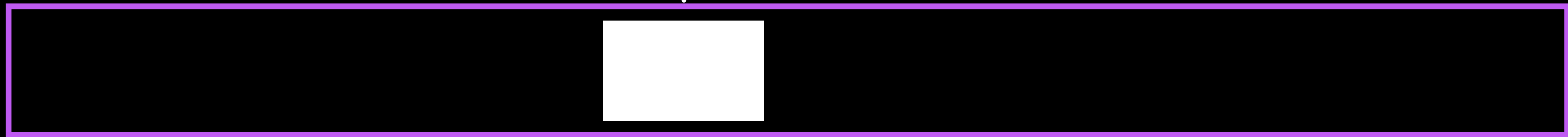
i-node table (on disk)



*i-node
(in-memory)*



buffer cache (in memory)



kernel space

user space

process

`fdatasync(fd)`

`open(path, O_DSYNC)`

All writes implicitly
call `fdatasync()`



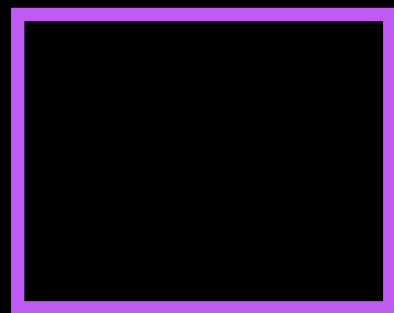
data blocks (on disk)



i-node table (on disk)



*i-node
(in-memory)*



buffer cache (in memory)



kernel space

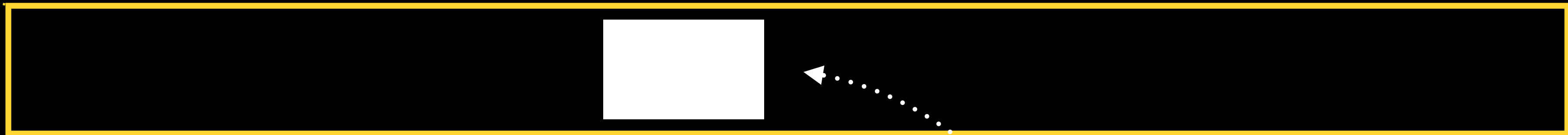
user space

process

```
fd = open(path, O_DIRECT)
```



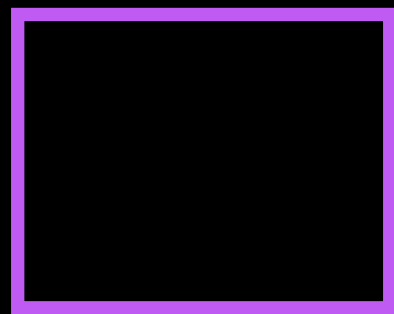
data blocks (on disk)



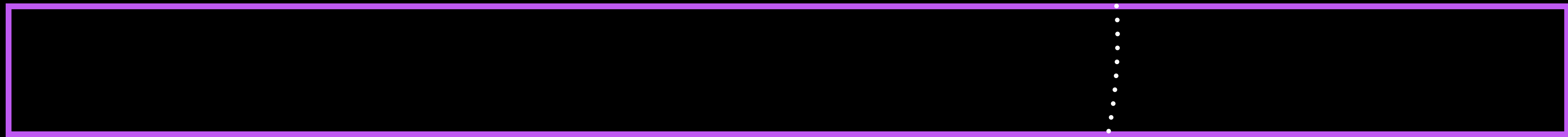
i-node table (on disk)



*i-node
(in-memory)*



buffer cache (in memory)



kernel space

user space

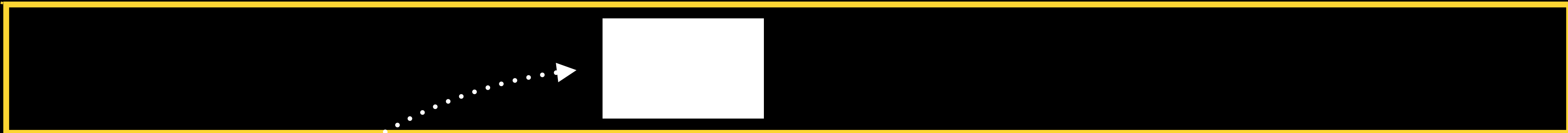
process

```
fd = open(path, O_DIRECT)
write(fd, data, count) ...
```

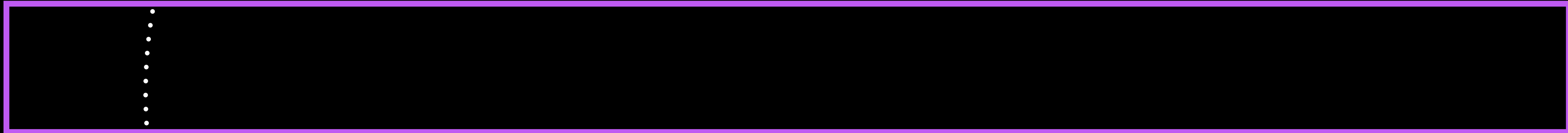
*Asynchronous
(write syscall may return before data hits disk)*



data blocks (on disk)



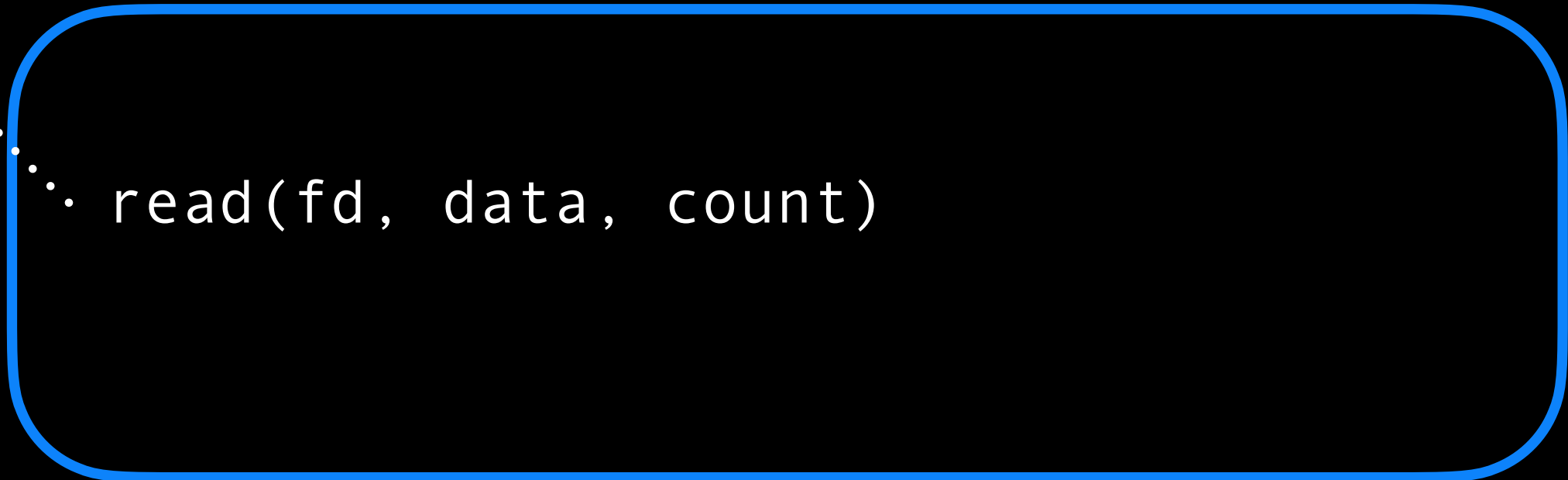
buffer cache (in memory)

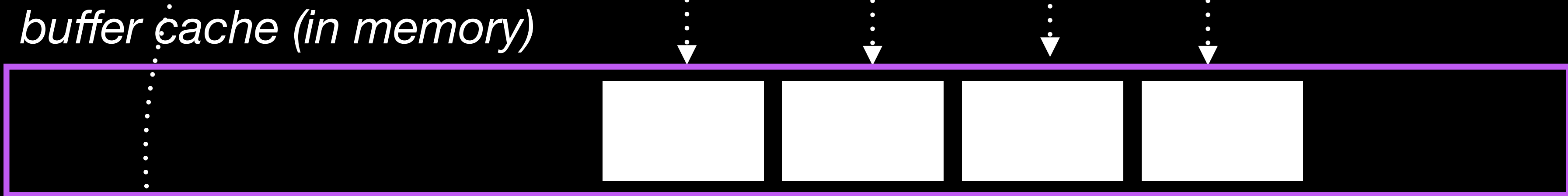


kernel space

user space

process

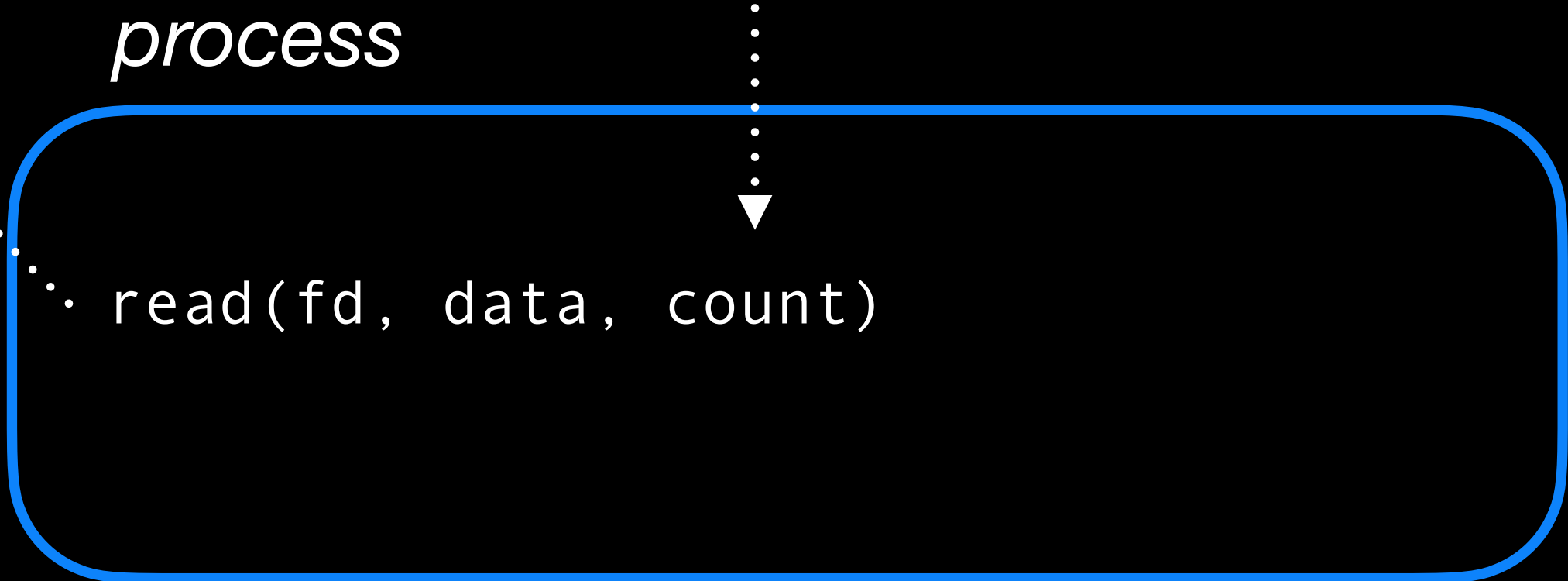




read-ahead

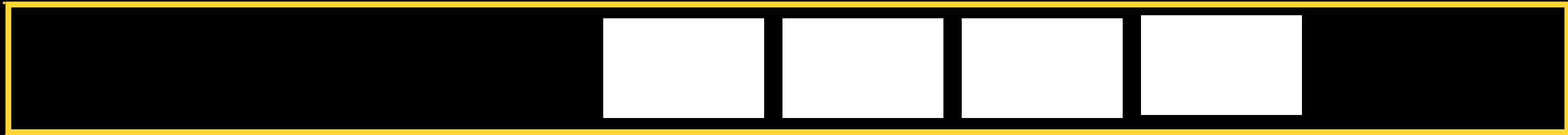
kernel space

user space

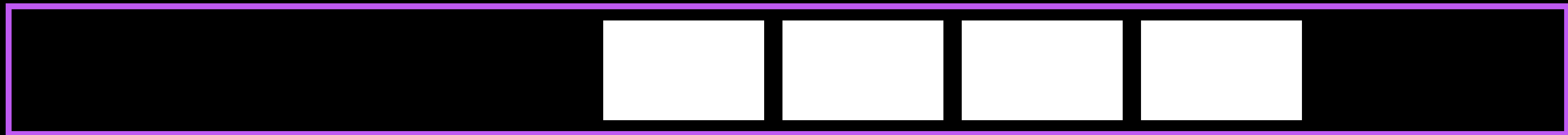




data blocks (on disk)



buffer cache (in memory)



kernel space

user space

process

```
read(fd, data, count)
read(fd, data, count)
```

see `posix_fadvise()` for specifying other access patterns

stdio I/O

stdio I/O

```
FILE *fp = fopen(path, mode)
```

stdio I/O

```
FILE *fp = fopen(path, mode)
```

```
FILE {
```

```
    fd
```

```
    buf
```



```
}
```

stdio I/O

```
FILE *fp = fopen(path, mode)
```

```
fwrite(data, size, nmemb, fp)
```

```
FILE {
```

```
    fd
```

```
    buf
```



```
}
```

stdio I/O

```
FILE *fp = fopen(path, mode)
```

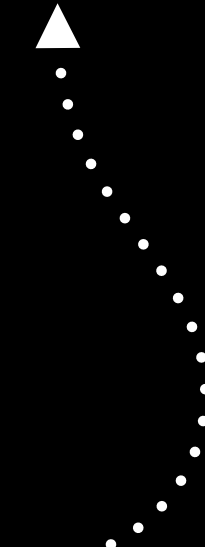
```
fwrite(data, size, nmemb, fp)
```

```
fwrite(data, size, nmemb, fp)
```

```
FILE {  
    fd  
    buf  
}
```

```
write(fd, buf) ...
```

kernel



stdio I/O

```
FILE *fp = fopen(path, mode)
```

```
FILE {
```

```
    fd
```

```
    buf
```



```
}
```

stdio I/O

```
FILE *fp = fopen(path, mode)
```

```
fwrite(data, size, nmemb, fp)
```

```
FILE {
```

```
    fd
```

```
    buf
```



```
}
```

stdio I/O

```
FILE *fp = fopen(path, mode)
```

```
fwrite(data, size, nmemb, fp)
```

```
fflush(fp)
```

```
FILE {
```

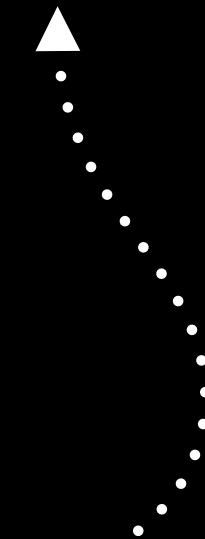
```
    fd
```

```
    buf
```

```
}
```

```
write(fd, buf) ...
```

kernel



stdio I/O

```
FILE *fp = fopen(path, mode)
```

```
FILE {
```

```
    fd
```

```
    buf
```



```
}
```

stdio I/O

```
FILE *fp = fopen(path, mode)
```

```
setvbuf(fp, NULL, _IOLBF, 0)
```

```
FILE {
```

```
    fd
```

```
    buf
```



```
}
```

stdio I/O

```
FILE *fp = fopen(path, mode)
```

```
setvbuf(fp, NULL, _IOLBF, 0)
```

```
fwrite(data, size, nmemb, fp)
```

```
FILE {
```

```
    fd
```

```
    buf
```

```
}
```



stdio I/O

```
FILE *fp = fopen(path, mode)
```

```
setvbuf(fp, NULL, _IOLBF, 0)
```

```
fwrite(data, size, nmemb, fp)
```

```
fwrite(data, size, nmemb, fp)
```

```
FILE {
```

```
    fd
```

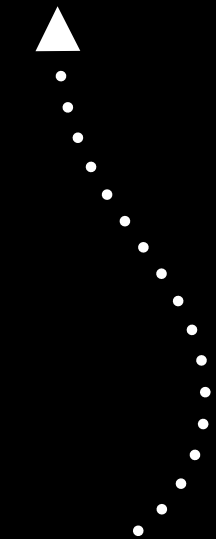
```
    buf
```

```
}
```

```
write(fd, buf) ...
```



kernel



system call	stdio library call
open	fopen
close	fclose
lseek	fseek, fsetpos, rewind, ftell, fgetpos
read	fread scanf, fscanf getchar, getc, fgetc, fgets, getline
write	fwrite, printf, fprintf putchar, putc, fputc, puts, fputs