

I have uploaded parser.c which parses the input pulling out for each event the process id and event that causes the state change.

You can use this code or write your own parser if my method and logic does not make sense to you. The reason it is included is so that you focus on the OS part of the exercises and not C parsing since this is the part that I received most questions about in Spring 2019.

In the code I parsed the file and wrote it to another file, you can of course, simply use the information storing it in variables rather than writing to a file. If you want to pull out the device name as well, you can comment **line 60** and uncomment **line 61**.

The general algorithm is

```
for each line in the file
    separate into events
    for the event
        separate using spaces
        pull out the process id and event word
    end for
end for
```

For example, if the line in the input file is

At time 5: P3 requests the disk; P7 is dispatched

LineInFile is an array of strings that will contain

LineInFile	At time 5	P3 requests the disk	P7 is dispatched
	[0]	[1]	[2]

Each event (LineInFile[1] and LineInFile[2]) are broken up by spaces in turn using the for loop

tokenizedLine	P3	requests	the	disk
	[0]	[1]	[2]	[3]

tokenizedLine	P7	is	dispatched
	[0]	[1]	[2]

So that you can pull out P3 requests P7 dispatched etc.