



IIT PALAKKAD

Indian Institute of Technology Palakkad
Department of Computer Science and Engineering
Operating Systems - Jul to Nov 2020
21 October 2020

Instructions:

- *You should do the assignment in the same group as you did lab4 and lab5*
- *Create a branch named "lab6" in the same repository you created previously*
- *Do the assignment, commit and push your changes when you are done.*
- *Submission deadline: 01-Nov, 2020 23:59 hrs.*

1. In this assignment, you have to trace how the state of a process changes over its lifetime. To do this, print a trace message on the console everytime the state of the process changes. The trace message should have the following information:
 - **pid** of the process
 - State transition **curr_state** → **next_state**
 - Reason for change if known statically (by looking at the code alone)

Therefore, a sample trace message will look as shown below:

```
xv6: fork(): pid 3 - UNUSED → EMBRYO  
xv6: fork(): pid 3 - EMBRYO → RUNNABLE
```

NOTE: You have to deal with the transition ZOMBIE to UNUSED differently. You have to mention the pid of the process that reaps another process, as well as the pid of the ZOMBIE process that is being reaped. Here, we wish to observe ZOMBIEs that are reaped by the init process. A sample output for this kind of trace message is as follows:

```
xv6: fork(): pid 3 - ZOMBIE → UNUSED (reaped by pid 2)
```

CAUTION: If you do not deal with locks carefully in your code, it is very likely that your xv6 kernel code will `panic()`. A simple backtrace of calling addresses will be printed along with the panic, that should help you debug your code.

Finally, test your code by porting a CPU-bound and an I/O-bound application from the modified UnixBench benchmarks that is uploaded on Moodle.