**Codespace with Process Run:**

[**https://classroom.github.com/a/ATQNduj9.**](https://classroom.github.com/a/ATQNduj9.)

**Instructions**  
Please run process-run.py with the commands listed below. Use the terminal for this exercise in Github Codespaces. You will need to change the execute permissions for process-run.py. For each **numbered** question, provide:

* A **numeric** or **yes/no** answer (brief).
* A **1–2 sentence justification** referring to the trace details (-c or -p). Submit your answers directly in Canvas using the **text entry** field, and clearly label each numbered response.

1. CPU Utilization (6 points)

**Command**: ./process-run.py -l 5:100,5:100

* (a) **Numeric (2 pts)**: What is the CPU utilization (in %)?
* (b) **Justification (4 pts)**: Why does this number make sense?

2. Completion Time (6 points)

**Command**: ./process-run.py -l 4:100,1:0

* (a) **Numeric (2 pts)**: How many time units does it take to finish both processes?
* (b) **Justification (4 pts)**: Provide one specific trace detail that confirms your answer.

3. Process Order (8 points)

Compare the outputs from:

1. ./process-run.py -l 4:100,1:0
2. ./process-run.py -l 1:0,4:100

* (a) **Yes/No (3 pts)**: Does reversing the order change total completion time?
* (b) **Justification (5 pts)**: Cite at least one reason from the trace explaining why (or why not).

4. Switching Policies (16 points total)

(a) SWITCH ON END (8 points)

**Command**: ./process-run.py -l 1:0,4:100 -S SWITCH\_ON\_END -c

* (a1) **Yes/No (2 pts)**: Does the scheduler switch away from a process performing I/O before it completes?
* (a2) **Justification (6 pts)**: In 1–2 sentences, explain how SWITCH\_ON\_END enforces that behavior.

(b) SWITCH ON IO (8 points)

**Command**: ./process-run.py -l 1:0,4:100 -S SWITCH\_ON\_IO -c

* (b1) **Yes/No (2 pts)**: Does the scheduler switch processes as soon as one issues an I/O request?
* (b2) **Justification (6 pts)**: Contrast SWITCH\_ON\_IO with SWITCH\_ON\_END in 1–2 sentences.

5. I/O Completion Policies (12 points total)

(a) IO\_RUN\_LATER (6 points)

**Command**:

./process-run.py -l 3:0,5:100,5:100,5:100 \ -S SWITCH\_ON\_IO -I IO\_RUN\_LATER -c -p

* (a1) **Yes/No (2 pts)**: Does the CPU become idle while a process that completed I/O remains unscheduled?
* (a2) **Justification (4 pts)**: How does this affect CPU utilization?

(b) IO\_RUN\_IMMEDIATE (6 points)

**Command**:

./process-run.py -l 3:0,5:100,5:100,5:100 \ -S SWITCH\_ON\_IO -I IO\_RUN\_IMMEDIATE -c -p

* (b1) **Yes/No (2 pts)**: Does the just-finished I/O process run immediately?
* (b2) **Justification (4 pts)**: Why might this improve overall performance?