



Exercício Prático – Simular, em linguagem de programação, o funcionamento do algoritmo Robin Round
Prof. Dra. Larissa Barbosa Leôncio Pinheiro

Considere os processos listados abaixo com seus respectivos tempos de chegada e execução. Implemente, em Python, uma simulação do algoritmo de Round Robin, com quantum igual a 2.

Processo	Chegada	Execução
P1	0	5
P2	1	3
P3	2	6

Instruções:

1. **Representação dos processos:** Crie uma estrutura (como uma lista de dicionários) para armazenar os processos com os seguintes dados:
 - Nome do processo
 - Tempo de chegada
 - Tempo de execução restante
 - Tempo em que começa a ser executado pela primeira vez
 - Tempo em que finaliza
2. **Simulação da passagem do tempo:** Implemente uma lógica que simule o andamento do tempo, verificando a cada instante se há processos que chegaram. Apenas processos cujo tempo de chegada seja menor ou igual ao tempo atual podem ser inseridos na fila de prontos.
3. **Execução com Round Robin:** A cada ciclo de 2 unidades de tempo (quantum), execute o processo que estiver na frente da fila:
 - Se ele finalizar dentro do quantum, registre o tempo de término
 - Se não finalizar, atualize o tempo restante e reintroduza o processo ao final da fila
 - Registre o tempo de início de execução apenas na primeira vez em que o processo é escalado.
4. **Resultados Esperados:** Ao final da simulação, exiba:
 - a) A ordem de execução (ex.: P1 → P2 → P3 → P1...)
 - b) O tempo de resposta de cada processo (tempo de início – chegada)
 - c) O tempo médio de resposta dos processos
 - d) Compare os valores obtidos com os resultados fornecidos pelo simulador: [Operating System Scheduling by LotusOregano](#)

Dica:

```
processos = [
    {"nome": "P1", "chegada": 0, "execucao": 5},
    {"nome": "P2", "chegada": 1, "execucao": 3},
    {"nome": "P3", "chegada": 2, "execucao": 6}
]
```

Utilize variáveis como *tempo*, *fila*, *ordem_execucao* e *resposta* para organizar os dados durante a simulação.