

Week 3! Assignment



Job Scheduler

You are tasked with implementing an advanced job scheduler that supports dynamic resource allocation, complex dependencies between jobs, prioritization based on job importance, and handling of job interruptions. Each job has a specific execution time, requires a resource, and may have dependencies that must be resolved before it can start.

Requirements:

1. Write a function `job_scheduler(jobs)` that takes a list of jobs.
2. Each job is represented by a tuple: `(job_name, execution_time_in_seconds, required_resources, dependencies, importance)`.
 - `job_name`: Name of the job.
 - `execution_time_in_seconds`: Time required to execute the job.
 - `required_resources`: List of resources needed to execute the job (e.g., `["CPU", "GPU"]`).
 - `dependencies`: List of job names that must finish before this job can start.
 - `importance`: Importance level of the job (higher number means higher priority).
3. Implement concurrent execution considering dynamic resource allocation and job dependencies.
4. Print the job name, start time (in seconds from the start of the scheduler), and the resources it used for each job.

Constraints:

1. Jobs requiring the same set of resources cannot execute concurrently.
2. Jobs can only start once all their dependencies have been satisfied.
3. Prioritize job execution based on importance level (higher number first).
4. If a higher priority job arrives while a lower priority job is executing, interrupt the lower priority job and resume it after the higher priority job finishes.

```

jobs = [
    ("Job1", 4, ["CPU"], [], 3),
    ("Job2", 2, ["GPU"], [], 2),
    ("Job3", 3, ["CPU", "GPU"], [], 4),
    ("Job4", 1, ["CPU"], ["Job1"], 1),
    ("Job5", 3, ["GPU"], ["Job2"], 3),
    ("Job6", 2, ["CPU", "GPU"], ["Job4"], 2)
]

job_scheduler(jobs)

```

```

Job1 started at 0 seconds using ['CPU']
Job2 started at 0 seconds using ['GPU']
Job3 started at 4 seconds using ['CPU', 'GPU']
Job5 started at 6 seconds using ['GPU']
Job4 started at 7 seconds using ['CPU']
Job6 started at 8 seconds using ['CPU', 'GPU']

```

Also test you code for this list,

```

jobs = [
    ("Job1", 6, ["CPU"], [], 3),
    ("Job2", 4, ["GPU"], [], 2),
    ("Job3", 8, ["CPU", "GPU"], [], 4),
    ("Job4", 3, ["CPU"], ["Job1"], 1),
    ("Job5", 5, ["GPU"], ["Job2"], 3),
    ("Job6", 7, ["CPU", "GPU"], ["Job4"], 2),
    ("Job7", 2, ["CPU"], [], 5),
    ("Job8", 4, ["GPU"], [], 3),
    ("Job9", 6, ["CPU", "GPU"], ["Job7", "Job8"], 2),
    ("Job10", 3, ["CPU"], ["Job1", "Job7"], 1),
    ("Job11", 5, ["GPU"], ["Job2", "Job8"], 3),
    ("Job12", 4, ["CPU"], ["Job10"], 2),
    ("Job13", 6, ["GPU"], ["Job5"], 4),
    ("Job14", 3, ["CPU", "GPU"], ["Job12", "Job13"], 1),
    ("Job15", 7, ["CPU"], ["Job3", "Job6"], 3),
    ("Job16", 5, ["GPU"], ["Job3", "Job9"], 2),
    ("Job17", 4, ["CPU", "GPU"], ["Job11", "Job14"], 4),
    ("Job18", 3, ["CPU"], ["Job10", "Job12"], 1),
    ("Job19", 6, ["GPU"], ["Job13", "Job16"], 3),

```

```
("Job20", 2, ["CPU", "GPU"], ["Job17", "Job18"], 2) ]
```

```
job_scheduler(jobs)
```