```
import logging
import os

from clusterops import (
    execute_with_cpu_cores,
    list_available_cpus,
)
from swarm_models import OpenAIChat

from swarms import Agent

# Configure logging
logging.basicConfig(level=logging.INFO)

api_key = os.getenv("OPENAI_API_KEY")

# Create an instance of the OpenAIChat class
model = OpenAIChat(
    openai_api_key=api_key,
    model_name="gpt-4o-mini",
    temperature=0.1,
    max_tokens=2000,
)



# Function for the director agent
```

```
def director_task(task: str):
    logging.info(f"Running Director agent for task: {task}")
    director = Agent(
        agent_name="Director",
        system_prompt="Directs the tasks for the workers",
        llm=model,
        max_loops=1,
        dashboard=False,
        streaming_on=True,
        verbose=True,
        stopping_token="<DONE>",
        state_save_file_type="json",
        saved_state_path="director.json",
    )
    return director.run(task)



# Function for worker 1
def worker1_task(task: str):
    logging.info(f"Running Worker1 agent for task: {task}")
    worker1 = Agent(
        agent_name="Worker1",
        system_prompt="Generates a transcript for a youtube video on what swarms are",
        llm=model,
        max_loops=1,
        dashboard=False,
```

```
        streaming_on=True,
        verbose=True,
        stopping_token="<DONE>",
        state_save_file_type="json",
        saved_state_path="worker1.json",
    )
    return worker1.run(task)


# Function for worker 2
def worker2_task(task: str):
    logging.info(f"Running Worker2 agent for task: {task}")
    worker2 = Agent(
        agent_name="Worker2",
        system_prompt="Summarizes the transcript generated by Worker1",
        llm=model,
        max_loops=1,
        dashboard=False,
        streaming_on=True,
        verbose=True,
        stopping_token="<DONE>",
        state_save_file_type="json",
        saved_state_path="worker2.json",
    )
    return worker2.run(task)
```

```
# CPU Core Assignment Example
def assign_tasks_to_cpus():
    # List available CPUs
    cpus = list_available_cpus()
    logging.info(f"Available CPUs: {cpus}")

    # Example: Assign Director task to 1 CPU core
    logging.info("Executing Director task using 1 CPU core")
    execute_with_cpu_cores(
        1, director_task, "Direct the creation of swarm video format"
    )

    # Example: Assign Worker1 task to 2 CPU cores
    logging.info("Executing Worker1 task using 2 CPU cores")
    execute_with_cpu_cores(
        2,
        worker1_task,
        "Generate transcript for youtube video on swarms",
    )

    # Example: Assign Worker2 task to 2 CPU cores
    logging.info("Executing Worker2 task using 2 CPU cores")
    execute_with_cpu_cores(
        2,
        worker2_task,
```

```
        "Summarize the transcript generated by Worker1",
    )


    print("finished")



if __name__ == "__main__":
    logging.info(
        "Starting the CPU-based task assignment for agents..."
    )
    assign_tasks_to_cpus()
```