

## Практикалық жұмыс 6

**Param-Bilim суперкомпьютерінде MPI технологияларымен жоғары өнімді есептеулер. Хабарды жіберу интерфейсі (MPI) дегеніміз не? Қарапайым параллель MPI бағдарламасының мысалы.**

Ауа-райының 2 сағаттағы өзгерісі

```
import time
import requests
import matplotlib.pyplot as plt
from datetime import datetime
import os

# ===== Параметрлер =====
api_key = "00100c661904f3dd546bef7c22c9b34a" # ☞ Мұнда өз API кілтіңізді қойыңыз!
city = "Astana,kz" # Қаланы өзгертуге болады (мысалы: "Almaty,kz")
duration_seconds = 2 * 60 * 60 # 2 сағат
interval = 600 # Әр 10 минут сайын (600 секунд)
save_dir = "weather_results" # Нәтижелер сақталатын қалта
os.makedirs(save_dir, exist_ok=True)
log_filename = os.path.join(save_dir, "weather_log.txt")
# =====

# ◇ Функция — ауа райын алу
def get_weather(api_key, city):
    base_url = "http://api.openweathermap.org/data/2.5/weather"
    params = {
        "q": city,
        "appid": api_key,
        "units": "metric"
    }
    resp = requests.get(base_url, params=params)
    data = resp.json()

    # Қате болса, ескерту
    if "main" not in data:
        print("⚠ Қате жауап алынды:", data)
        return None, None, None

    temp = data["main"]["temp"]
    humidity = data["main"]["humidity"]
    wind = data["wind"]["speed"]
    return temp, humidity, wind

# ◇ Лог файлын дайындау
with open(log_filename, "w", encoding="utf-8") as f:
    f.write("time,temp,humidity,wind\n")
```

```

# ◇ Мәліметтер тізімдері
times = []
temps = []
humidities = []
winds = []

start_time = time.time()
print("☁️ Аяу райын бақылау басталды...\n")

try:
    while True:
        elapsed = time.time() - start_time
        if elapsed >= duration_seconds:
            break

        # Аяу райын алу
        temp, humidity, wind = get_weather(api_key, city)
        now = datetime.now().strftime("%H:%M:%S")

        if temp is None:
            print("🕒 Дерек келмеді, 10 минуттан кейін қайта тексереміз...")
            time.sleep(interval)
            continue

        # Мәліметтерді сақтау
        times.append(now)
        temps.append(temp)
        humidities.append(humidity)
        winds.append(wind)

        with open(log_filename, "a", encoding="utf-8") as f:
            f.write(f'{now},{temp},{humidity},{wind}\n')

        print(f'[{now}] 🌡️ {temp}°C | 💧 {humidity}% | 🌬️ {wind} м/с")

        # 10 минут күту
        time.sleep(interval)

    print("\n✅ 2 сағаттық бақылау аяқталды. График сызылып жатыр...")

except KeyboardInterrupt:
    print("\n🛑 Пайдаланушы тоқтатты. График дайындалады...")

# ◇ График салу
plt.figure(figsize=(10, 6))
plt.plot(times, temps, label="Температура (°C)", marker="o")
plt.plot(times, humidities, label="Ылғалдылық (%)", marker="x")
plt.plot(times, winds, label="Жел жылдамдығы (м/с)", marker="s")
plt.xticks(rotation=45)

```

```
plt.xlabel("Уақыт")
plt.ylabel("Мәндер")
plt.title(f"Ауа райының 2 сағаттағы өзгерісі — {city}")
plt.legend()
plt.tight_layout()

# ♦ Графикті сақтау
graph_path = os.path.join(save_dir, "weather_plot.png")
plt.savefig(graph_path)
print("📄 График сақталды:", graph_path)

plt.show()
```

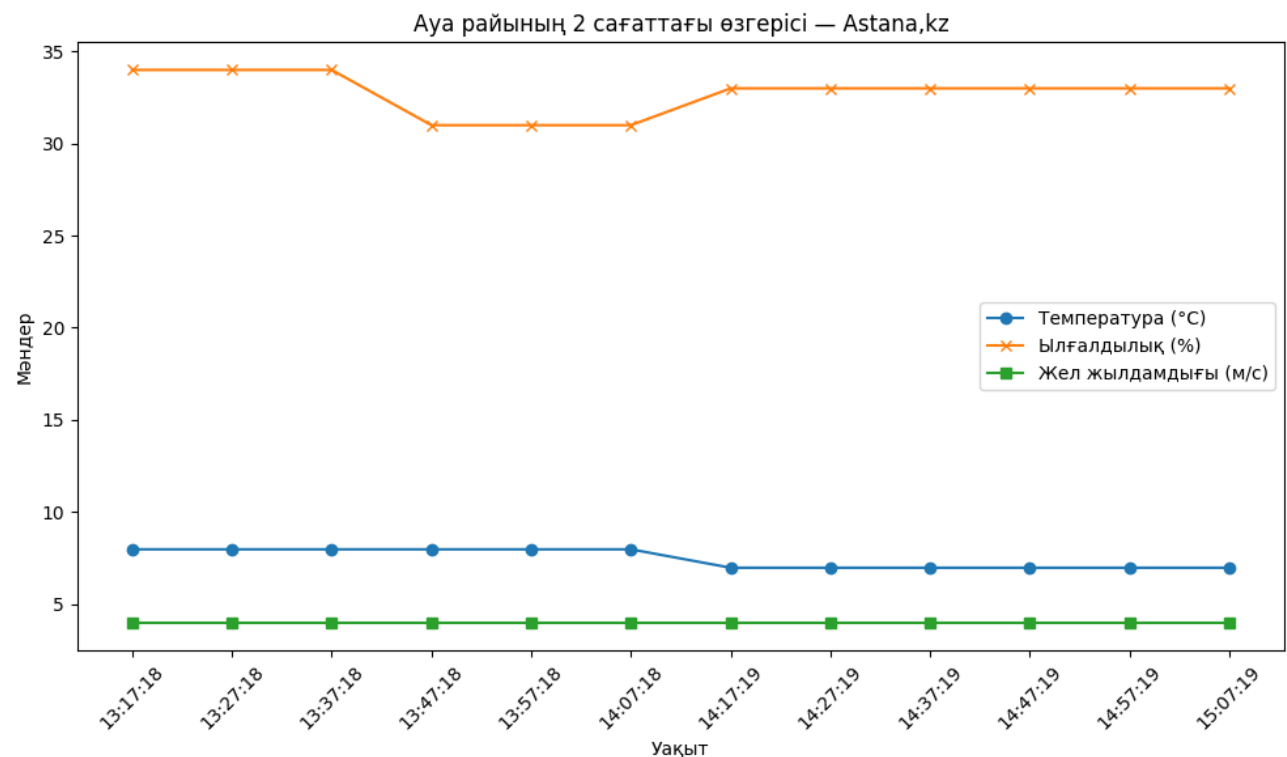
Нәтиже:

🌤️ Ауа райын бақылау басталды...

|            |           |  |       |  |          |
|------------|-----------|--|-------|--|----------|
| [13:17:18] | 🌡️ 7.97°C |  | 💧 34% |  | 🌬️ 4 м/с |
| [13:27:18] | 🌡️ 7.97°C |  | 💧 34% |  | 🌬️ 4 м/с |
| [13:37:18] | 🌡️ 7.97°C |  | 💧 34% |  | 🌬️ 4 м/с |
| [13:47:18] | 🌡️ 7.97°C |  | 💧 31% |  | 🌬️ 4 м/с |
| [13:57:18] | 🌡️ 7.97°C |  | 💧 31% |  | 🌬️ 4 м/с |
| [14:07:18] | 🌡️ 7.97°C |  | 💧 31% |  | 🌬️ 4 м/с |
| [14:17:19] | 🌡️ 6.97°C |  | 💧 33% |  | 🌬️ 4 м/с |
| [14:27:19] | 🌡️ 6.97°C |  | 💧 33% |  | 🌬️ 4 м/с |
| [14:37:19] | 🌡️ 6.97°C |  | 💧 33% |  | 🌬️ 4 м/с |
| [14:47:19] | 🌡️ 6.97°C |  | 💧 33% |  | 🌬️ 4 м/с |
| [14:57:19] | 🌡️ 6.97°C |  | 💧 33% |  | 🌬️ 4 м/с |
| [15:07:19] | 🌡️ 6.97°C |  | 💧 33% |  | 🌬️ 4 м/с |

✅ 2 сағаттық бақылау аяқталды. График сызылып жатыр...

📄 График сақталды: weather\_results/weather\_plot.png



**Тапсырма:**

Осы программаны MPI негізінде есептеңіз.