МИКРОПРОЦЕССОРЛЫҚ ЖҮЙЕЛЕР

8-ДӘРІС

Процестер, ағындар және нақты уақыттағы басқару

әл-Фараби атындағы Қазақ ұлттық университеті

Физика-техникалық факультет

ОҚУ МАҚСАТЫ

Дәрістің мақсаты және оқу нәтижелері

Мақсат: процестер, ағындар және нақты уақыттағы басқару бойынша теориялық ұғымдарды жүйелеу және Raspberry Pi 5 негізінде тәжірибелік талдау жүргізу.

- Негізгі архитектуралық және бағдарламалық ұғымдарды түсіндіру
- Raspberry Pi 5 жүйесінде тиісті құралдарды қолдану
- Инженерлік шешімді қауіпсіздік пен өнімділік тұрғысынан бағалау

ҚҰРЫЛЫМ

Дәріс жоспары

01 Процесс және ағын

Процесс оқшауланған ресурстарға, ағын ортақ жадыға ие.

02 Жоспарлаушы

Linux тапсырмаларды процессор ядроларына бөледі.

03 Кідіріс пен jitter

Жауап беру уақыты өзгермелі болуы мүмкін.

04 Python threading

I/O бағытталған міндеттерде ағындарды қолдану.

05 asyncio

Оқиғаға негізделген асинхронды бағдарламалау моделі.

06 Нақты уақыт шектеулері

Стандартты Linux қатаң real-time кепілдік бермейді.

Процесс және ағын

Процесс оқшауланған ресурстарға, ағын ортақ жадыға ие.

Инженерлік талдау

Бұл компонент жүйенің қандай функциясын орындайды?

Жоспарлаушы

Linux тапсырмаларды процессор ядроларына бөледі.

Инженерлік талдау

Өнімділікке, кідіріске немесе сенімділікке әсерін түсіндіріңіз.

Кідіріс пен jitter

Жауап беру уақыты өзгермелі болуы мүмкін.

Инженерлік талдау

Қай жағдайда басқа технологияны таңдаған дұрыс?

Python threading

I/O бағытталған міндеттерде ағындарды қолдану.

Инженерлік талдау

Бұл компонент жүйенің қандай функциясын орындайды?

asyncio

Оқиғаға негізделген асинхронды бағдарламалау моделі.

Инженерлік талдау

Өнімділікке, кідіріске немесе сенімділікке әсерін түсіндіріңіз.

Нақты уақыт шектеулері

Стандартты Linux қатаң real-time кепілдік бермейді.

Инженерлік талдау

Қай жағдайда басқа технологияны таңдаған дұрыс?

Функционалдық схема және компоненттердің байланысы

Жоспарлау

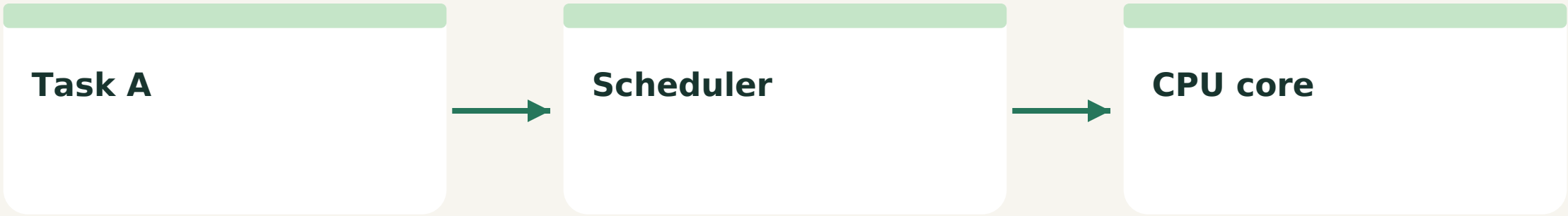

Схема — оқыту мақсатындағы функционалдық модель; нақты электрлік қосылу сызбасы емес.

Практикалық мысал 1

Терминал / Python 3

```
import threading
print(threading.active_count())
```

Орындау алдында қажетті пакеттер мен құрылғылардың қолжетімділігін тексеріңіз.

Практикалық мысал 2

Терминал / Python 3

```
import asyncio
async def main():
    await asyncio.sleep(0.1)
asyncio.run(main())
```

Орындау алдында қажетті пакеттер мен құрылғылардың қолжетімділігін тексеріңіз.

Практикалық мысал 3

Терминал / Python 3

```
ps -eo pid,comm,pri,ni | head
```

Орындау алдында қажетті пакеттер мен құрылғылардың қолжетімділігін тексеріңіз.