

№6 Дәріс. Хабар алмасу модельдері: Message Passing, RPC, RMI

Дәріс мақсаты:

- Үлестірілген жүйелерде процестердің өзара әрекеттесу тәсілдерін түсіндіру;
- Message Passing, Remote Procedure Call (RPC) және Remote Method Invocation (RMI) модельдерінің принциптерін қарастыру;
- Әр модельдің артықшылықтары мен кемшіліктерін талдау;
- Үлестірілген қосымшаларды құруда қолданылатын коммуникация механизмдеріне мысалдар келтіру.

Кілт сөздер:

Хабар алмасу, Message Passing, RPC, RMI, клиент–сервер, байланыс арнасы, marshaling, stub, skeleton, synchronous, asynchronous communication, distributed communication, remote invocation.

Дәріс жоспары

1. Үлестірілген жүйелердегі коммуникация ұғымы
2. Message Passing моделі
 - Ұғымы
 - Түрлері
 - Артықшылықтары / кемшіліктері
 - Қолдану салалары
3. Remote Procedure Call (RPC)
 - RPC принципі
 - Stub, marshaling
 - Жұмыс схемасы
 - Артықшылықтары / кемшіліктері
4. Remote Method Invocation (RMI)
 - RMI ерекшелігі
 - Объект бағытталған шақыру
 - RMI архитектурасы
5. Хабар алмасу модельдерін салыстыру
6. Қорытынды

Кіріспе

1. Үлестірілген жүйелердегі коммуникация

Үлестірілген жүйеде процестер мен түйіндер бір-бірімен ақпарат алмасып, ортақ мақсатты орындайды. Коммуникацияның негізгі түрлері:

Түрі	Сипаттамасы
Тікелей	Процестер бір-біріне тікелей жібереді
Жанама	Қойма (message queue) арқылы
Синхронды	Жауап күту

Түрі	Сипаттамасы
Асинхронды	Хабар жіберген соң жалғастыру

2. Message Passing — хабар алмасу моделі

Анықтама:

Процестер хабарламалар жіберу және қабылдау арқылы өзара әрекеттеседі.

Негізгі операциялар:

SCSS

```
send(message, destination)
```

```
receive(message, source)
```

Түрлері:

Түрі	Түсінігі
Синхронды	Жіберуші қабылдаушы дайын болғанша күтеді
Асинхронды	Хабарды жіберіп, жұмысын жалғастырады
Буферленген	Хабарлар уақытша сақталуы мүмкін
Тікелей	Адрес тікелей көрсетіледі
Жанама	Queue / mailbox арқылы

Артықшылықтары:

- Жылдам, қарапайым
- Реал-тайм жүйелер үшін қолайлы
- Shared memory қажет емес

Кемшіліктері:

- Программалауда күрделі процесс басқару
- Хабар жоғалуы мүмкін
- Синхрондау қиын

Қолдану:

- MPI (Message Passing Interface) — суперкомпьютерлерде
- ZeroMQ, RabbitMQ, Kafka сияқты жүйелер

3. RPC — Remote Procedure Call

Анықтама:

Қашықтағы машинадағы функцияны жергілікті функциядай шақыру.

Негізгі идеясы:

Жергілікті процедура шақырғандай, қашықтағы процедураға сұраныс жіберу

Архитектура компоненттері:

Компонент	Мәні
Client Stub	Параметрді marshaling жасайды
Server Stub (Skeleton)	Параметрді қабылдайды, функцияны шақырады
Marshaling	Параметрлерді желіге бейімдеу

Компонент	Мәні
Unmarshaling	Деректерді қалпына келтіру

Жұмыс схемасы:

1. Клиент функцияны шақырады
2. Клиент stub параметрлерді сериализациялайды
3. Хабар серверге барады
4. Сервер stub функцияны орындайды
5. Нәтиже кері marshaling
6. Клиент нәтижені алады

Артықшылықтары:

- Жазу жеңіл, түсінікті
- Процестер арасында жоғары деңгейлі API

Кемшіліктері:

- Желіге тәуелді
- Қателерді өңдеу күрделі
- Платформалық үйлесімділік қиын (Big endian vs little endian)

Протоколдар / технологиялар:

- gRPC
- ONC RPC
- SOAP / XML-RPC
- JSON-RPC

4.RMI — Remote Method Invocation

*Анықтама:*Java ортасында объект әдістерін қашықтан шақыру механизмі.

Ерекшелігі:

RPC-ден айырмашылығы:

RMI объектілермен, RPC процедуралармен жұмыс істейді

Архитектура:

- Remote Interface
- Stub
- Skeleton
- RMI Registry

Артықшылықтары:

- Java үшін native шешім
- Объект–бағытталған архитектура

Кемшіліктері:

- Тек Java
- Кешігулер көп болуы мүмкін

5.Модельдерді салыстыру

Ерекшелік	Message Passing	RPC	RMI
Абстракция деңгейі	Төмен	Орта	Жоғары

Ерекшелік	Message Passing	RPC	RMI
Коммуникация	Хабар алмасу	Процедура	Объект әдісі
Синхрондылық	Бар/жоқ	Әдетте синхронды	Синхронды
Жүйе тәуелділігі	Жоқ	Иә	Java
Қолданылуы	HPC, real-time	Web services	Java distributed apps

Қорытынды

Таңдау жүйе мақсатына байланысты:

- **Message Passing** — HPC, IoT, real-time
- **RPC** — серверлік архитектура, микросервистер
- **RMI** — Java enterprise, J2EE

Бақылау сұрақтары

1. Message Passing моделінің негізгі идеясы қандай?
2. Синхронды және асинхронды хабар алмасудың айырмашылығы неде?
3. RPC жұмыс істеу принципін түсіндіріңіз.
4. Stub және skeleton деген не?
5. RMI қандай жағдайда тиімді?
6. RPC мен RMI арасындағы негізгі айырмашылық қандай?

Пайдаланылған әдебиеттер

1. Tanenbaum A. *Distributed Systems: Principles and Paradigms*
2. Coulouris G. *Distributed Systems: Concepts and Design*
3. Birrell & Nelson. *Implementing Remote Procedure Calls*
4. ORACLE Java RMI Documentation
5. gRPC Official Documentation