

## Wywoływanie procedur zdalnych — RPC

Zagadnienia projektowe  
Zagadnienia realizacyjne

---

---

---

---

---

---

---

---

### Mechanizm wywołania procedury

```
main(int argc, char* argv[]){  
    int id, status;  
    id = atoi(argv[1]);  
    status = zabij_proc(id);  
    exit(status);  
}  
:  
:  
int zabij_proces (int pid){  
    int stat;  
    stat = kill(pid, 9);  
    return stat;  
}
```

RPC

2

---

---

---

---

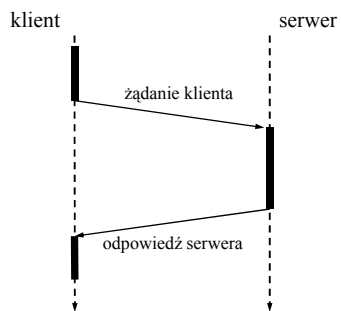
---

---

---

---

### Schemat interakcji klient-serwer (transakcja komunikatu)



RPC

3

---

---

---

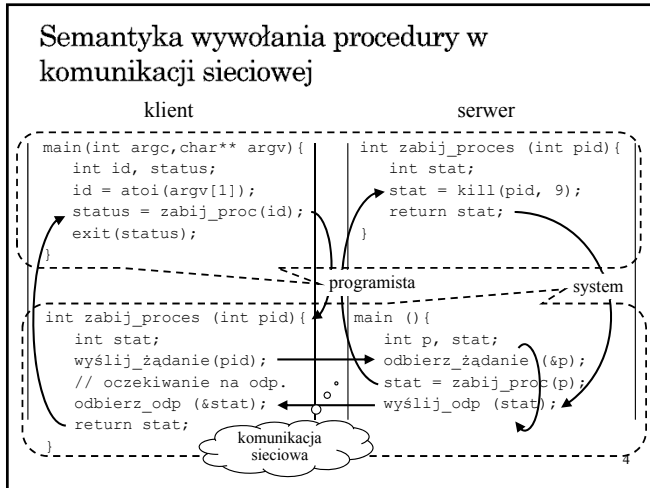
---

---

---

---

---




---

---

---

---

---

---

---

---

---

---

### Zagadnienia projektowe

- przezroczystość dostępu — ukrycie komunikacji sieciowej przed aplikacją przez odpowiednie opakowanie funkcji komunikacyjnych namiastką klienta oraz serwera.
- gwarancja wykonania — ukrywanie błędów komunikacyjnych
- specyfikacja interfejsu — sposób opisu sygnatur procedur zdalnych (nazwy, typy parametrów)
- obsługa sytuacji wyjątkowych

RPC 5

---

---

---

---

---

---

---

---

---

---

### Przezroczystość dostępu

- namiastka klienta (ang. client stub) — udostępnienie aplikacji klienckiej procedury lokalnej odpowiedzialnej za przesłanie danych do serwera oraz odebranie wyników
- namiastka serwera (ang. server stub) — udostępnienie aplikacji po stronie serwera procedury lokalnej odpowiedzialnej za odebranie identyfikatora procedury zdalnej do wywołania, parametrów procedury, a odesłanie wyników lub zgłoszenie wyjątków

RPC 6

---

---

---

---

---

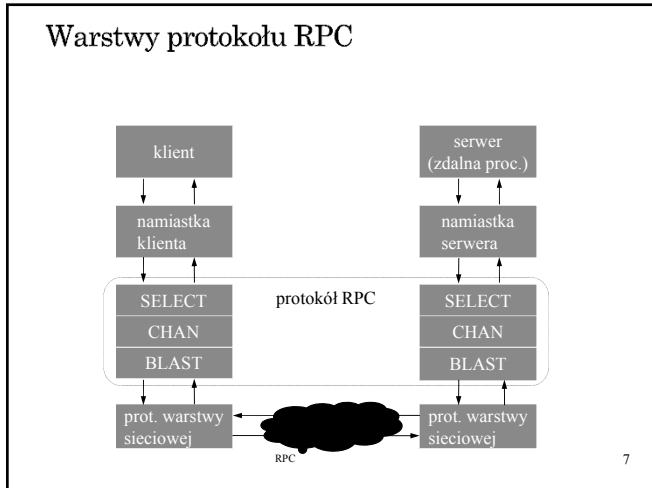
---

---

---

---

---




---

---

---

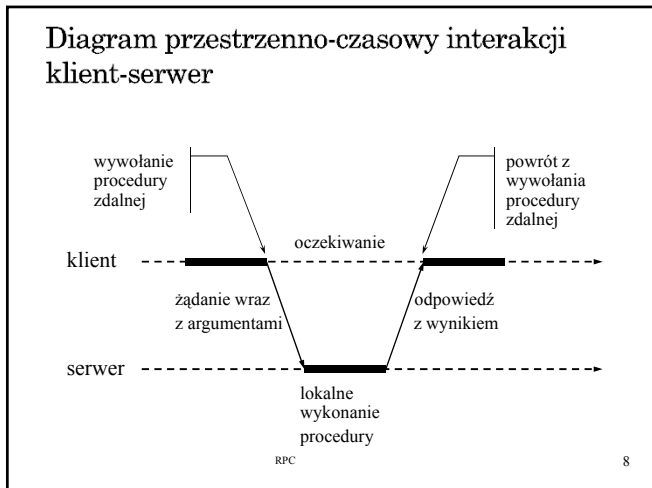
---

---

---

---

---




---

---

---

---

---

---

---

---

- ### Przekazywanie parametrów
- przekazywanie przez wartość (ang. call-by-value)
    - ↳ problem różnic w reprezentacji danych, np. w kodowaniu znaków, uporządkowaniu bajtów, w formatach liczb zmiennopozycyjnych itp.
  - przekazywanie przez referencje (ang. call-by-reference)
    - ↳ problem zinterpretowania wartości wskaźnika w innej przestrzeni adresowej
  - przekazywanie przez kopiowanie i odtwarzanie —
    - ↳ problem przetaczania (kwestia opisu struktur danych w celu prawidłowego zidentyfikowania wszystkich składowych)
- 9

---

---

---

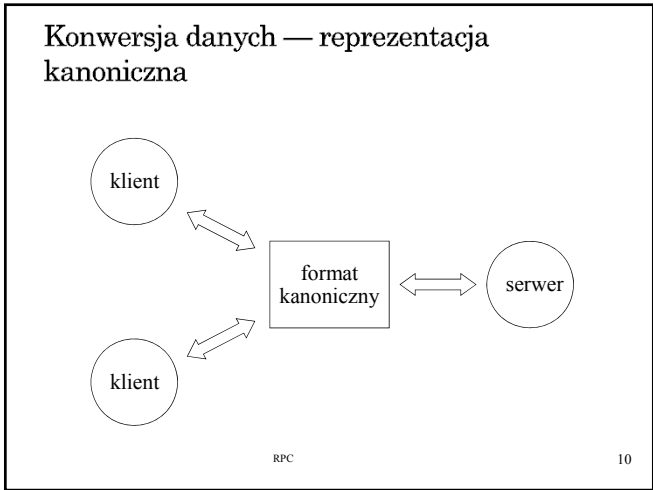
---

---

---

---

---



---

---

---

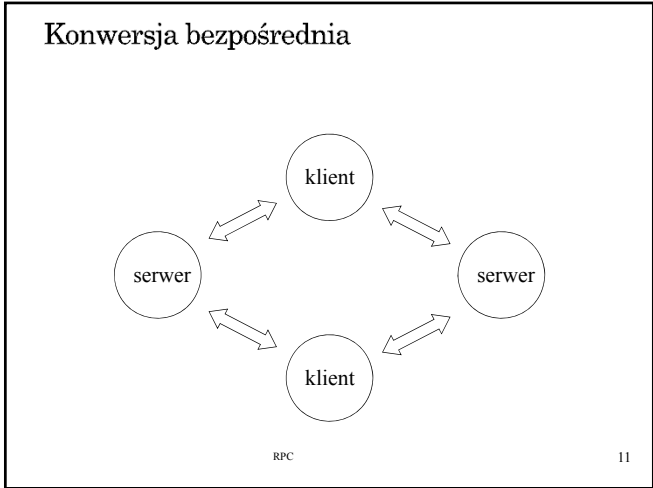
---

---

---

---

---



---

---

---

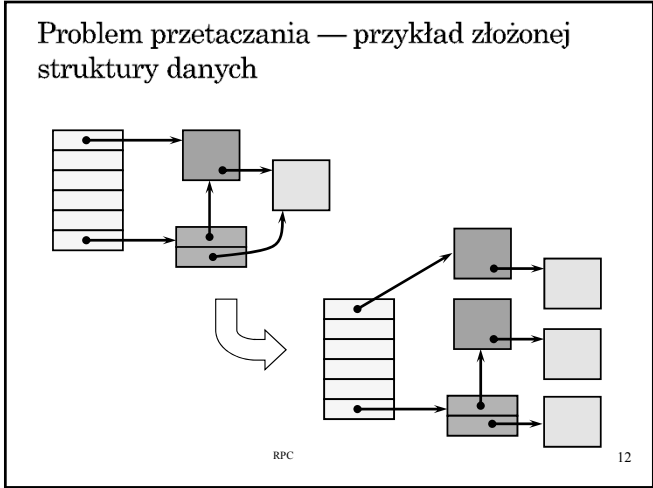
---

---

---

---

---



---

---

---

---

---

---

---

---

### Gwarancja wykonania (semantyka błędu)

- Semantyka *co najmniej raz* — po uzyskaniu odpowiedzi (wyniku) od serwera klient ma pewność, że wywoływana procedura wykonała się co najmniej raz.
- Semantyka *co najwyżej raz* — po uzyskaniu odpowiedzi (wyniku) od serwera klient wie, że wywoływana procedura wykonała się dokładnie raz.

#### Jak należy zinterpretować przypadek wystąpienia błędu (wyjątku) w wywołaniu procedury zdalnej?

- Semantyka *dokładnie raz* — niemożliwa do uzyskania, jeśli system narażony jest na awarie (np. serwera lub łączy).
- Semantyka *ewentualnie* — brak gwarancji, procedura mogła się wykonać lub mogła się nie wykonać.

RPC

13

---

---

---

---

---

---

---

---

### Zagadnienia realizacyjne

- Przetwarzanie interfejsu
- Wiązanie klienta z serwerem
- Obsługa komunikacji klient-serwer
- Realizacja semantyki błędu
- Problem osieroconych obliczeń

RPC

14

---

---

---

---

---

---

---

---

### Przetwarzanie interfejsu procedur zdalnych

- Generowanie namiastki klienta
- Generowanie namiastki serwera
- Generowanie przykładowego programu klienta (client sample)
- Generowanie wzorca do implementacji procedur zdalnych (template)
- Generowanie plików do zarządzania kompilacją

RPC

15

---

---

---

---

---

---

---

---

### Wiązanie klienta z serwerem

- Wiązanie statyczne — klient ma wprowadzony na stałe identyfikator komunikacyjny serwera (np. para: adres IP, nr portu).
- Wiązanie dynamiczne — klient uzyskuje adres serwera za pośrednictwem łącznika (np. portmap lub rpcbind w Sun RPC).

RPC

16

---

---

---

---

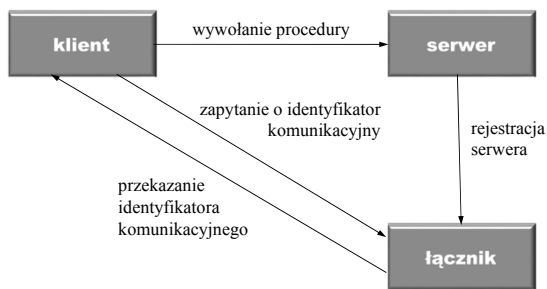
---

---

---

---

### Wiązanie dynamiczne klienta z serwerem



RPC

17

---

---

---

---

---

---

---

---

### Obsługa komunikacji klient-serwer

- BLAST — realizuje przesyłanie dużych komunikatów poprzez podział na mniejsze części, transmisję poszczególnych części i ponowne złożenie w jeden komunikat po stronie odbiorczej,
- CHAN — synchronizuje wymianę komunikatów z żądaniami wywołania procedur oraz odpowiedziami,
- SELECT — rozdziela i przekazuje komunikaty z żądaniami do odpowiednich procesów.

RPC

18

---

---

---

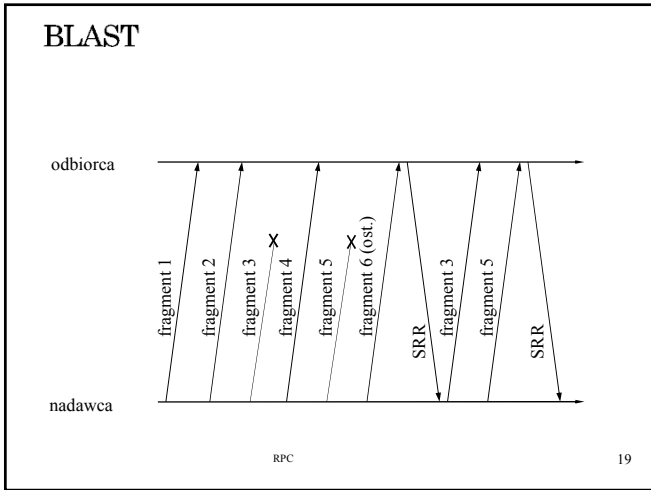
---

---

---

---

---



---

---

---

---

---

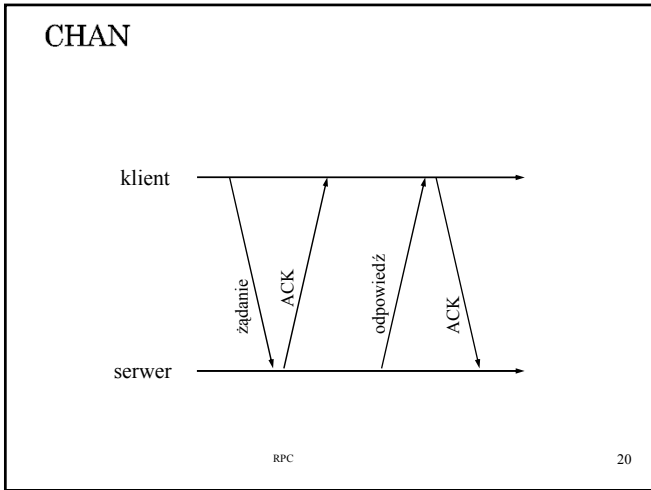
---

---

---

---

---



---

---

---

---

---

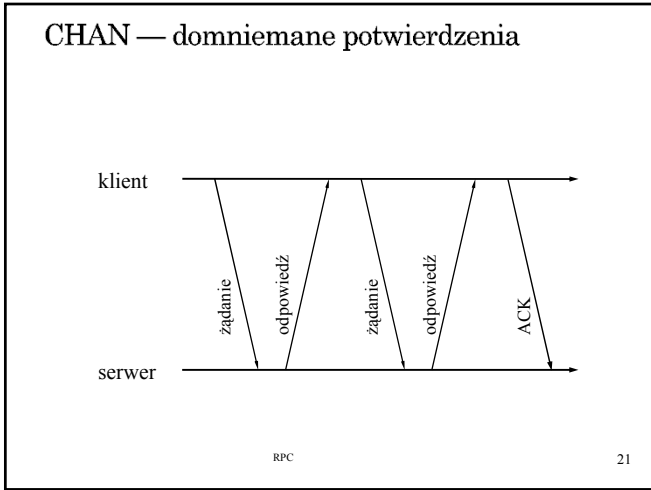
---

---

---

---

---



---

---

---

---

---

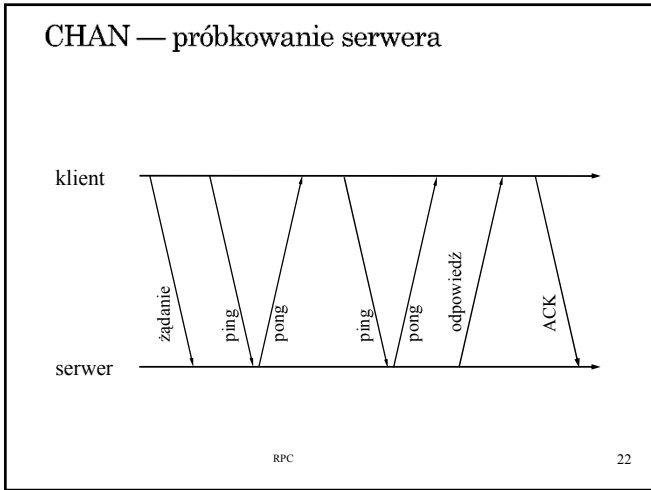
---

---

---

---

---



---

---

---

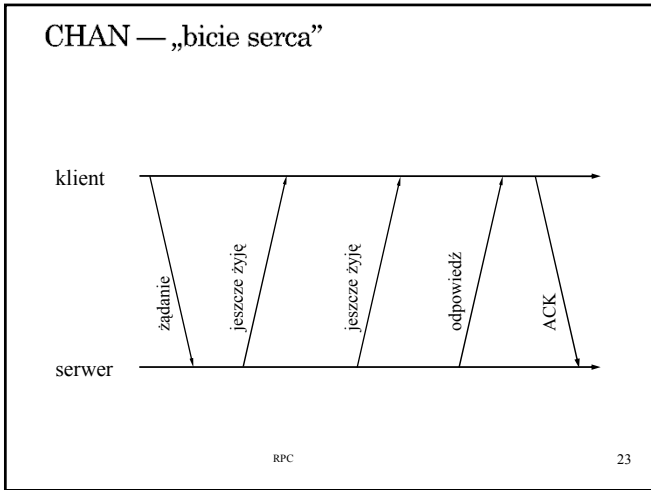
---

---

---

---

---



---

---

---

---

---

---

---

---

- ### Realizacja semantyki błędu
- nie można zlokalizować serwera — zgłoszenie wyjątku
  - zaginione żądanie — retransmisja żądania po upływie ustalonego czasu oczekiwania
  - zaginiona odpowiedź — retransmisja żądania
    - ↳ stosowanie procedur idempotentnych
    - ↳ numerowanie żądań i retransmisja odpowiedzi
  - awaria serwera
    - ↳ przed podjęciem realizacji → retransmisja żądania
    - ↳ po wykonaniu → zgłoszenie wyjątku
  - awaria klienta — osierocenie obliczeń
- RPC 24

---

---

---

---

---

---

---

---



### Usuwanie osieroconych obliczeń (1)

- Eksterminacja — rejestrowanie działań podejmowanych przez klienta na nośniku niewrażliwym na awarie i usuwanie na tej podstawie osieroconych obliczeń po restarcie klienta.
- Reinkarnacja — każdy restart klienta rozpoczyna nową epokę (identyfikowaną przez numer kolejny), po której usuwane są wszystkie obliczenia związane z poprzednią epoką.

RPC

25

---

---

---

---

---

---

---

---

### Usuwanie osieroconych obliczeń (2)

- Łagodna reinkarnacja — reinkarnacja, w której usuwa się tylko te obliczenia rozpoczęte w starej epoce, dla których nie ma właściciela.
- Wygaśnięcie — przydział określonego czasu  $T$  serwerowi na wykonanie procedury. Jeśli wykonanie nie zakończy się w czasie  $T$ , serwer musi uzyskać kolejny przydział, pod warunkiem, że obliczenia nie zostały osierocone. Jeśli klient odczeka czas  $T$  przy restarcie, osierocone obliczenia same się zakończą.

RPC

26

---

---

---

---

---

---

---

---

### SELECT

- po stronie klienta: odwzorowanie wywoływanej procedury na jej identyfikator, przekazywany do serwera.
- po stronie serwera: zlokalizowanie wywoływanej procedury na podstawie identyfikatora.

RPC

27

---

---

---

---

---

---

---

---

### Warianty użycia mechanizmu RPC

- Wywołanie asynchroniczne — klient nie czeka na wynik wykonania procedury zdalnej (wykonywanie procedury zdalnej odbywa się równoległe z przetwarzaniem po stronie klienta).
- Wywołanie zwrotne — klient udostępnia procedurę zdalną po swojej stronie i przekazuje serwerowi informacje umożliwiające jej wywołanie (zamiana ról pomiędzy klientem a serwerem).

RPC

28

---

---

---

---

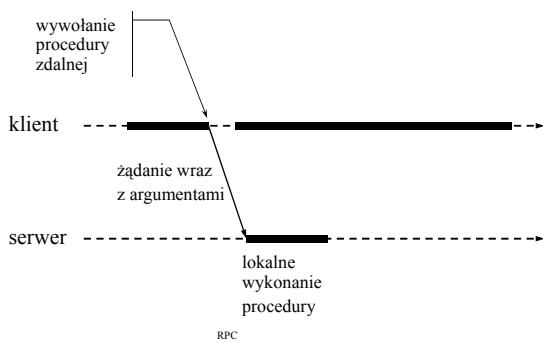
---

---

---

---

### Wywołanie asynchroniczne



RPC

29

---

---

---

---

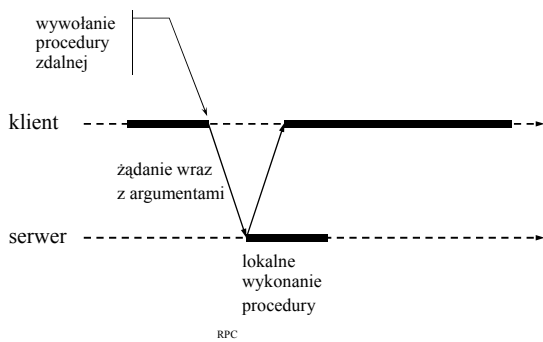
---

---

---

---

### Wywołanie asynchroniczne z potwierdzeniem odbioru



RPC

30

---

---

---

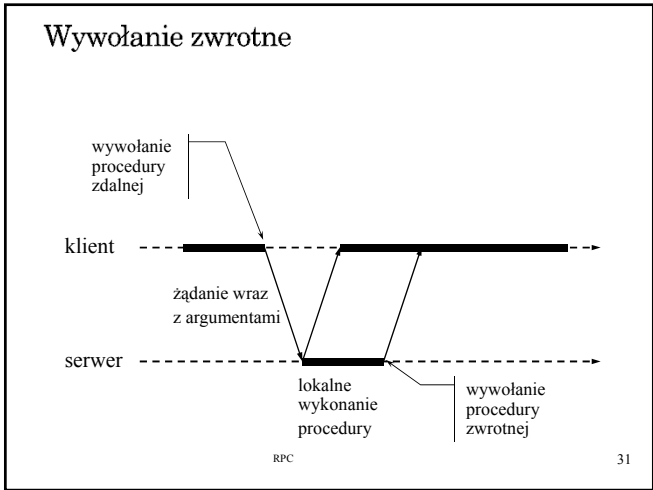
---

---

---

---

---



---

---

---

---

---

---

---

---