Programowanie wspóªbie»ne Wykªad kursowy dla II roku informatyki

Marcin Engel

Instytut Informatyki

Uniwersytet Warszawski

rok akademicki 2016/17

Program wspóªbie»ny

Praca wspóªbie»na polega na tym, »e skªadaj¡ce si¦ na ni¡ zjawiska, czynno±ci lub dziaªania odbywaj¡ si¦ równocze±nie. Istotny jest przy tym punkt widzenia obserwatora (. . . )

Program wspóªbie»ny

Praca wspóªbie»na polega na tym, »e skªadaj¡ce si¦ na ni¡ zjawiska, czynno±ci lub dziaªania odbywaj¡ si¦ równocze±nie. Istotny jest przy tym punkt widzenia obserwatora (. . . )

Wykonanie wspóªbie»ne polega na tym, »e kolejna czynno±¢ rozpoczyna si¦ przed zako«czeniem poprzedniej.

Program wspóªbie»ny

Praca wspóªbie»na polega na tym, »e skªadaj¡ce si¦ na ni¡ zjawiska, czynno±ci lub dziaªania odbywaj¡ si¦ równocze±nie. Istotny jest przy tym punkt widzenia obserwatora (. . . )

Wykonanie wspóªbie»ne polega na tym, »e kolejna czynno±¢ rozpoczyna si¦ przed zako«czeniem poprzedniej.

Program wspóªbie»ny jest zbiorem programów sekwencyjnych, które mog¡ by¢ wykonywane równolegle.

Dlaczego rozwa»amy programy wspóªbie»ne?

1 Niektóre problemy s¡ z natury wspóªbie»ne, rozwi¡zania innych daj¡ si¦ ªatwo i elegancko wyrazi¢ w postaci niezale»nie wykonuj¡cych si¦ procedur

Dlaczego rozwa»amy programy wspóªbie»ne?

1 Niektóre problemy s¡ z natury wspóªbie»ne, rozwi¡zania innych daj¡ si¦ ªatwo i elegancko wyrazi¢ w postaci niezale»nie wykonuj¡cych si¦ procedur

2 Systemy operacyjne optymalizuj¡ czas obrotu zadania wykonuj¡c inny proces, gdy pewien proces musi poczeka¢ na zako«czenie operacji wej±cia-wyj±cia

Dlaczego rozwa»amy programy wspóªbie»ne?

1 Niektóre problemy s¡ z natury wspóªbie»ne, rozwi¡zania innych daj¡ si¦ ªatwo i elegancko wyrazi¢ w postaci niezale»nie wykonuj¡cych si¦ procedur

2 Systemy operacyjne optymalizuj¡ czas obrotu zadania wykonuj¡c inny proces, gdy pewien proces musi poczeka¢ na zako«czenie operacji wej±cia-wyj±cia

3 Stale zwi¦kszaj¡ca si¦ moc obliczeniowa procesorów umo»liwia wykonanie wielu zada« na jednym procesorze w sposób niezauwa»alny dla u»ytkownika

Dlaczego rozwa»amy programy wspóªbie»ne?

1 Niektóre problemy s¡ z natury wspóªbie»ne, rozwi¡zania innych daj¡ si¦ ªatwo i elegancko wyrazi¢ w postaci niezale»nie wykonuj¡cych si¦ procedur

2 Systemy operacyjne optymalizuj¡ czas obrotu zadania wykonuj¡c inny proces, gdy pewien proces musi poczeka¢ na zako«czenie operacji wej±cia-wyj±cia

3 Stale zwi¦kszaj¡ca si¦ moc obliczeniowa procesorów umo»liwia wykonanie wielu zada« na jednym procesorze w sposób niezauwa»alny dla u»ytkownika

4 Malej¡ce ceny sprz¦tu sprzyjaj¡ powstawaniu architektur wieloprocesorowych

Dlaczego rozwa»amy programy wspóªbie»ne?

1 Niektóre problemy s¡ z natury wspóªbie»ne, rozwi¡zania innych daj¡ si¦ ªatwo i elegancko wyrazi¢ w postaci niezale»nie wykonuj¡cych si¦ procedur

2 Systemy operacyjne optymalizuj¡ czas obrotu zadania wykonuj¡c inny proces, gdy pewien proces musi poczeka¢ na zako«czenie operacji wej±cia-wyj±cia

3 Stale zwi¦kszaj¡ca si¦ moc obliczeniowa procesorów umo»liwia wykonanie wielu zada« na jednym procesorze w sposób niezauwa»alny dla u»ytkownika

4 Malej¡ce ceny sprz¦tu sprzyjaj¡ powstawaniu architektur wieloprocesorowych

5 Rozpowszechnienie si¦ sieci komputerowych umo»liwia rozpraszanie oblicze«

Dlaczego rozwa»amy programy wspóªbie»ne?

1 Niektóre problemy s¡ z natury wspóªbie»ne, rozwi¡zania innych daj¡ si¦ ªatwo i elegancko wyrazi¢ w postaci niezale»nie wykonuj¡cych si¦ procedur

2 Systemy operacyjne optymalizuj¡ czas obrotu zadania wykonuj¡c inny proces, gdy pewien proces musi poczeka¢ na zako«czenie operacji wej±cia-wyj±cia

3 Stale zwi¦kszaj¡ca si¦ moc obliczeniowa procesorów umo»liwia wykonanie wielu zada« na jednym procesorze w sposób niezauwa»alny dla u»ytkownika

4 Malej¡ce ceny sprz¦tu sprzyjaj¡ powstawaniu architektur wieloprocesorowych

5 Rozpowszechnienie si¦ sieci komputerowych umo»liwia rozpraszanie oblicze«

6 U»ywaj¡c wielu procesorów mo»na wykona¢ program szybciej ni» na jednym.

Przykªad sortowanie przez scalanie

1 Niektóre problemy s¡ z natury wspóªbie»ne . . .

procedure Sortuj (i , j ) { i < j }

begin

{ ... }

m := (i+j) div 2;

sortuj ( i , m);

sortuj (m+1, j);

scalaj ( i , m, j)

end

Przykªad sortowanie przez scalanie

1 Niektóre problemy s¡ z natury wspóªbie»ne . . .

procedure Sortuj (i , j ) { i < j }

begin

{ ... }

m := (i+j) div 2;

cobegin

sortuj ( i , m);

sortuj (m+1, j);

coend

scalaj ( i , m, j)

end

Przykªad przetwarzanie potokowe

1 . . . rozwi¡zania innych daj¡ si¦ ªatwo i elegancko wyrazi¢ w postaci niezale»nie wykonuj¡cych si¦ procedur

procedure G;

var

x: integer;

begin

x := 2;

while true do begin

write (1, x );

x := x + 1

end

end;

Przykªad przetwarzanie potokowe

1 . . . rozwi¡zania innych daj¡ si¦ ªatwo i elegancko wyrazi¢ w postaci niezale»nie wykonuj¡cych si¦ procedur

var

x, r : integer;

begin

read (0, r );

while true do begin

read (0, x );

if x mod r <> 0 then write (1, x);

end

end;

cobegin G; F; F; ... coend

Dlaczego rozwa»amy programy wspóªbie»ne?

2 Systemy operacyjne optymalizuj¡ czas obrotu zadania wykonuj¡c inny proces, gdy pewien proces musi poczeka¢ na zako«czenie operacji wej±cia-wyj±cia

3 Stale zwi¦kszaj¡ca si¦ moc obliczeniowa procesorów umo»liwia wykonanie wielu zada« na jednym procesorze w sposób niezauwa»alny dla u»ytkownika

Dlaczego rozwa»amy programy wspóªbie»ne?

2 Systemy operacyjne optymalizuj¡ czas obrotu zadania wykonuj¡c inny proces, gdy pewien proces musi poczeka¢ na zako«czenie operacji wej±cia-wyj±cia

3 Stale zwi¦kszaj¡ca si¦ moc obliczeniowa procesorów umo»liwia wykonanie wielu zada« na jednym procesorze w sposób niezauwa»alny dla u»ytkownika

Wieloprogramowo±¢ i wielozadaniowo±¢

Dlaczego rozwa»amy programy wspóªbie»ne?

4 Malej¡ce ceny sprz¦tu sprzyjaj¡ powstawaniu architektur wieloprocesorowych

Dlaczego rozwa»amy programy wspóªbie»ne?

4 Malej¡ce ceny sprz¦tu sprzyjaj¡ powstawaniu architektur wieloprocesorowych

Intel Core i7: 4 rdzenie + HT = 8 procesorów wirtualnych. Cena ok. 1000 zª.

Dlaczego rozwa»amy programy wspóªbie»ne?

4 Malej¡ce ceny sprz¦tu sprzyjaj¡ powstawaniu architektur wieloprocesorowych

Intel Core i7: 4 rdzenie + HT = 8 procesorów wirtualnych. Cena ok. 1000 zª.

GeForce GTX 260: ponad 200 procesorów. Cena ok. 800 zª.

Dlaczego rozwa»amy programy wspóªbie»ne?

4 Malej¡ce ceny sprz¦tu sprzyjaj¡ powstawaniu architektur wieloprocesorowych

Intel Core i7: 4 rdzenie + HT = 8 procesorów wirtualnych. Cena ok. 1000 zª.

GeForce GTX 260: ponad 200 procesorów. Cena ok. 800 zª. Biblioteki (OpenCL, CUDA) pozwalaj¡ce wykorzystywa¢ do oblicze« procesory karty gra cznej.

Dlaczego rozwa»amy programy wspóªbie»ne?

5 Rozpowszechnienie si¦ sieci komputerowych umo»liwiªo rozpraszanie oblicze«

Dlaczego rozwa»amy programy wspóªbie»ne?

5 Rozpowszechnienie si¦ sieci komputerowych umo»liwiªo rozpraszanie oblicze«

Parallel Virtual Machine

Message Passing Interface

Systemy rozproszone

Grid computing

Cloud computing

SETI home i inne

Przyspieszenie programu równolegªego

6 U»ywaj¡c wielu procesorów mo»na wykona¢ program szybciej ni» na jednym.

Przyspieszenie programu równolegªego

6 U»ywaj¡c wielu procesorów mo»na wykona¢ program szybciej ni» na jednym.

Przyspieszenie: Stosunek czasu potrzebnego jednemu procesorowi na wykonanie zadania do czasu potrzebnego *n* równolegªym procesorom na wykonanie tego samego zadania.

Podstawowe poj¦cia

Uwaga!

Zjawiska, o których b¦dzie mowa, nie s¡ widoczne dla zwykªego u»ytkownika, a nawet dla programisty aplikacji.

Rozwa»amy rozkazy j¦zyka maszynowego, które ostatecznie wykonuj¡ procesy.

Podstawowe poj¦cia. Proces

Proces to wykonanie programu.

Proces jest obiektem aktywnym w odró»nieniu od (tekstu) programu.

W systemie z podziaªem czasu wykonuje si¦ wiele procesów. Niektóre procesy mog¡ wykonywa¢ ten sam program. Proces wykonuje jeden program . . .

Podstawowe poj¦cia. Proces

Proces to wykonanie programu.

Proces jest obiektem aktywnym w odró»nieniu od (tekstu) programu.

W systemie z podziaªem czasu wykonuje si¦ wiele procesów. Niektóre procesy mog¡ wykonywa¢ ten sam program. Proces wykonuje jeden program . . .

. . . ale mo»e zmieni¢ wykonywany program.

Skªadowe procesu

W poj¦ciu procesu mie±ci si¦:

przydzielona mu pami¦¢, czyli przestrze« adresowa

zbiór rejestrów

inne zasoby potrzebne do wykonywania programu

informacje administracyjne (np.: stan, kod zako«czenie . . . )

Skªadowe procesu. Przestrze« adresowa

zawiera kod programu (text section)

zawiera dane (data section)

zawiera stos procesu (stack section)

Skªadowe procesu. Przestrze« adresowa

zawiera kod programu (text section)

zawiera dane (data section)

zawiera stos procesu (stack section)

jest adresowana od zera do . . . (adresy logiczne)

jest sprz¦towo odwzorowywana na pami¦¢ zyczn¡ (translacja adresów)

nie musi w caªo±ci znajdowa¢ si¦ w pami¦ci zycznej (pami¦¢ wirtualna)

Skªadowe procesu. Przestrze« adresowa

zawiera kod programu (text section)

zawiera dane (data section)

zawiera stos procesu (stack section)

jest adresowana od zera do . . . (adresy logiczne)

jest sprz¦towo odwzorowywana na pami¦¢ zyczn¡ (translacja adresów)

nie musi w caªo±ci znajdowa¢ si¦ w pami¦ci zycznej (pami¦¢ wirtualna)

podlega ochronie przed innymi procesami

Skªadowe procesu. Zbiór rejestrów

W systemie z podziaªem czasu dziaªa wiele procesów. Wykonanie procesu jest przerywane przez ró»ne zdarzenia.

Skªadowe procesu. Zbiór rejestrów

W systemie z podziaªem czasu dziaªa wiele procesów. Wykonanie procesu jest przerywane przez ró»ne zdarzenia. Rejestr rozkazów, wska¹nik stosu, rejestry ogólnego

przeznaczenia, rejestr stanu oraz rejestry specjalne (np. do zarz¡dzania pami¦ci¡) okre±laj¡ przebieg wykonania procesu. Proces musi odtworzy¢ stan, w którym byª przed przerwaniem wykonania.

System operacyjny (by¢ mo»e z pomoc¡ sprz¦tu) musi zapami¦tywa¢ stan tych rejestrów przerywaj¡c wykonanie procesu.

Program wspóªbie»ny

Program wspóªbie»ny skªada si¦ ze sko«czonej liczby procesów sekwencyjnych.

Program wspóªbie»ny

Program wspóªbie»ny skªada si¦ ze sko«czonej liczby procesów sekwencyjnych.

Ka»dy proces sekwencyjny wykonuje ci¡g operacji atomowych.

Proces a program

Program wspóªbie»ny to zbiór procesów sekwencyjnych. Proces jest obiektem aktywnym w odró»nieniu od (tekstu) programu.

Proces sekwencyjny to wykonanie programu sekwencyjnego.

Mo»liwe wykonania programu wspóªbie»nego

1 Równolegªe synchroniczne

2 Równolegªe asynchroniczne

3 W przeplocie

Wykonanie równolegªe synchroniczne

Typowe dla architektur wielordzeniowych

Wspólny zegar taktuj¡cy

Wszystkie procesory wykonuj¡ w tym samym momencie kolejn¡ faz¦ procesora

Wykonanie równolegªe asynchroniczne

Typowe dla architektur rozproszonych

Brak wspólnego zegara

Ka»dy procesor pracuje we wªasnym tempie

Podziaª czasu

Ka»dy proces otrzymuje kwant czasu.

Podziaª czasu

Ka»dy proces otrzymuje kwant czasu.

Po jego wykorzystaniu procesor odkªada wykonanie procesu i rozpoczyna wykonanie innego procesu jest to przeª¡czenie kontekstu

Podziaª czasu

Ka»dy proces otrzymuje kwant czasu.

Po jego wykorzystaniu procesor odkªada wykonanie procesu i rozpoczyna wykonanie innego procesu jest to przeª¡czenie kontekstu

Proces mo»e nie wykorzysta¢ caªego kwantu czasu (gdy za»¡da operacji wej±cia-wyj±cia)

Przeª¡czenie kontekstu

Proces P System operacyjny Pr

Przeª¡czenie kontekstu

Proces P System operacyjny Proces Q

Przeª¡czenie kontekstu

Proces P System operacyjny Proces Q Przerwanie lub wywoªanie systemowe

Przeª¡czenie kontekstu

Proces P System operacyjny Proces Q

Przerwanie lub wywoªanie systemowe

Przechowaj stan w bloku kontrolnym P

Odtwórz stan z bloku kontrolnego Q

Przeª¡czenie kontekstu

Proces P System operacyjny Proces Q

Przerwanie lub wywoªanie systemowe

Przechowaj stan w bloku kontrolnym P

Odtwórz stan z bloku kontrolnego Q

Przeª¡czenie kontekstu

Proces P System operacyjny Proces Q

Przerwanie lub wywoªanie systemowe

Przechowaj stan w bloku kontrolnym P

Odtwórz stan z bloku kontrolnego Q

Przeª¡czenie kontekstu

Proces P System operacyjny Proces Q

Przerwanie lub wywoªanie systemowe

Przechowaj stan w bloku kontrolnym P

Odtwórz stan z bloku kontrolnego Q

Przerwanie lub wywoªanie systemowe

Przeª¡czenie kontekstu

Proces P System operacyjny Proces Q

Przerwanie lub wywoªanie systemowe

Przechowaj stan w bloku kontrolnym P

Odtwórz stan z bloku kontrolnego Q

Przerwanie lub wywoªanie systemowe

Przechowaj stan w bloku kontrolnym Q

Odtwórz stan z bloku kontrolnego P

Przeª¡czenie kontekstu

Proces P System operacyjny Proces Q

Przerwanie lub wywoªanie systemowe

Przechowaj stan w bloku kontrolnym P

Odtwórz stan z bloku kontrolnego Q

Przerwanie lub wywoªanie systemowe

Przechowaj stan w bloku kontrolnym Q

Odtwórz stan z bloku kontrolnego P

Przeª¡czenie kontekstu

Proces P System operacyjny Proces Q

Przerwanie lub wywoªanie systemowe

Przechowaj stan w bloku kontrolnym P

Odtwórz stan z bloku kontrolnego Q

Przerwanie lub wywoªanie systemowe

Przechowaj stan w bloku kontrolnym Q

Odtwórz stan z bloku kontrolnego P

Stany procesu Nowy

Stany procesu

Nowy

Gotowy

Stany procesu

Nowy

Gotowy Aktywny

Stany procesu

Nowy

Gotowy Aktywny

Stany procesu

Nowy

Gotowy Aktywny

Wstrzymany

Stany procesu

Nowy

Gotowy Aktywny

Wstrzymany

Stany procesu

Nowy

Gotowy Aktywny

Wstrzymany

Zako«czony

Kolejki procesów

System operacyjny umieszcza PCB procesów w kolejkach: kolejce (kolejkach) procesów gotowych

kolejkach procesów wstrzymanych na danym zasobie

Kolejka procesów

Pocz¡tek

gotowych PCB PCB PCB Koniec

Pocz¡tek

Dysk

Koniec

PCB PCB PCB

Wspóªbie»no±¢ a równolegªo±¢

Równolegªo±¢ faktycznie jednoczesne wykonywanie wielu czynno±ci

Wspóªbie»no±¢ a równolegªo±¢

Równolegªo±¢ faktycznie jednoczesne wykonywanie wielu czynno±ci

Wspóªbie»no±¢ abstrakcja równolegªo±ci. Wykonanie wspóªbie»ne mo»e odbywa¢ si¦ równolegªe lub w przeplocie.

Operacje atomowe

Ka»dy proces wykonuje ci¡g operacji atomowych

Operacja atomowa wykonuje si¦ od pocz¡tku do ko«ca, zanim rozpocznie si¦ wykonanie kolejnej operacji atomowej (tego samego lub innego procesu)

Operacje atomowe sa niepodzielne

Operacje atomowe

Ka»dy proces wykonuje ci¡g operacji atomowych

Operacja atomowa wykonuje si¦ od pocz¡tku do ko«ca, zanim rozpocznie si¦ wykonanie kolejnej operacji atomowej (tego samego lub innego procesu)

Operacje atomowe sa niepodzielne

Operacj¡ atomow¡ mo»e by¢ (w zale»no±ci od przyjmowanego modelu lub platformy sprz¦towej):

pojedyncza instrukcja j¦zyka wysokiego poziomu

pojedynczy rozkaz

pojedynczy cykl pracy procesora

Typowe operacje atomowe

Jednoprocesorowy system operacyjnych z niewywªaszczalnym j¡drem

Typowe operacje atomowe

Jednoprocesorowy system operacyjnych z niewywªaszczalnym j¡drem

gdy proces wykonuje si¦ w trybie j¡dra, za operacj¦ atomow¡ mo»na przyj¡¢ ci¡g instrukcji wykonywanych mi¦dzy dobrze okre±lonymi punktami wywªaszczania

Typowe operacje atomowe

Jednoprocesorowy system operacyjnych z niewywªaszczalnym j¡drem

gdy proces wykonuje si¦ w trybie j¡dra, za operacj¦ atomow¡ mo»na przyj¡¢ ci¡g instrukcji wykonywanych mi¦dzy dobrze okre±lonymi punktami wywªaszczania

gdy proces znajduje si¦ w trybie u»ytkownika, za operacj¦ atomow¡ mo»na uzna¢ pojedynczy rozkaz maszynowy

Typowe operacje atomowe

Jednoprocesorowy system operacyjnych z niewywªaszczalnym j¡drem

gdy proces wykonuje si¦ w trybie j¡dra, za operacj¦ atomow¡ mo»na przyj¡¢ ci¡g instrukcji wykonywanych mi¦dzy dobrze okre±lonymi punktami wywªaszczania

gdy proces znajduje si¦ w trybie u»ytkownika, za operacj¦

atomow¡ mo»na uzna¢ pojedynczy rozkaz maszynowy

System wieloprocesorowy nie gwarantuje niepodzielno±ci nawet pojedynczego rozkazu maszynowego

Typowe operacje atomowe

Jednoprocesorowy system operacyjnych z niewywªaszczalnym j¡drem

gdy proces wykonuje si¦ w trybie j¡dra, za operacj¦ atomow¡ mo»na przyj¡¢ ci¡g instrukcji wykonywanych mi¦dzy dobrze okre±lonymi punktami wywªaszczania

gdy proces znajduje si¦ w trybie u»ytkownika, za operacj¦

atomow¡ mo»na uzna¢ pojedynczy rozkaz maszynowy

System wieloprocesorowy nie gwarantuje niepodzielno±ci nawet pojedynczego rozkazu maszynowego

W j¦zykach modelowania u»ytkownik sam okre±la operacje atomowe

Operacje atomowe przykªad

int x;

x++;

Kompilator gcc kompiluje t¦ instrukcj¦ do jednego rozkazu maszynowego

Rozkaz ten jest niepodzielny w architekturach

jednoprocesorowych

Na wieloprocesorze mo»e doj±¢ do przeplotu faz

poszczególnych procesów i rozkaz przestaje by¢ niepodzielny Inne kompilatory (a nawet gcc z wyª¡czon¡ optymalizacj¡) mog¡ dokona¢ translacji w inny sposób, np. na kilka rozkazów

Wnioski

Nic wolno nic zakªada¢ o:

niepodzielno±ci instrukcji j¦zyka

Wnioski

Nic wolno nic zakªada¢ o:

niepodzielno±ci instrukcji j¦zyka

sposobie wykonywania programu wspóªbie»nego

Wnioski

Nic wolno nic zakªada¢ o:

niepodzielno±ci instrukcji j¦zyka

sposobie wykonywania programu wspóªbie»nego

platformie sprz¦towej, na której program b¦dzie wykonywany

Obliczenie programu wspóªbie»nego

Procesy sekwencyjne wykonuj¡ swoje instrukcje atomowe jedna po drugiej w kolejno±ci zgodnej z porz¡dkiem programu

Obliczenie programu wspóªbie»nego

Procesy sekwencyjne wykonuj¡ swoje instrukcje atomowe jedna po drugiej w kolejno±ci zgodnej z porz¡dkiem programu Ka»dy przeplot instrukcji atomowych poszczególnych procesów zgodny z ich porz¡dkiem programu nazwiemy obliczeniem lub scenariuszem wykonania

Obliczenie programu wspóªbie»nego

Procesy sekwencyjne wykonuj¡ swoje instrukcje atomowe jedna po drugiej w kolejno±ci zgodnej z porz¡dkiem programu Ka»dy przeplot instrukcji atomowych poszczególnych procesów zgodny z ich porz¡dkiem programu nazwiemy obliczeniem lub scenariuszem wykonania

Zakªadamy zatem, »e na skutek wykonania programu

wspóªbie»nego mo»e pojawi¢ si¦ dowolne obliczenie, tzn. kolejna wykonywana instrukcja atomowa pochodzi z dowolnego procesu

Czy taka abstrakcja ma sens?

Przecie»:

W systemie z podziaªem czasu przeª¡czenie kontekstu nie pojawia si¦ po ka»dym kolejnym rozkazie

Czy taka abstrakcja ma sens?

Przecie»:

W systemie z podziaªem czasu przeª¡czenie kontekstu nie pojawia si¦ po ka»dym kolejnym rozkazie

W systemach wieloprocesorowych wykonanie programu nie jest przeplotem operacji, ale ich faktycznym równolegªym

wykonaniem

Czy taka abstrakcja ma sens?

Przecie»:

W systemie z podziaªem czasu przeª¡czenie kontekstu nie pojawia si¦ po ka»dym kolejnym rozkazie

W systemach wieloprocesorowych wykonanie programu nie jest przeplotem operacji, ale ich faktycznym równolegªym

wykonaniem

W systemach rozproszonych poszczególne w¦zªy s¡ niezale»ne i ich dziaªanie nie jest nawet koordynowane wspólnym zegarem

Uzasadnienie abstrakcji

W systemie z podziaªem czasu:

nie sposób z góry przewidzie¢, kiedy nast¡pi przeª¡czenie kontekstu

przyj¦cie zaªo»enia co do liczby wykonywanych w jednym kwancie rozkazów wprowadziªoby zale»no±ci od konkretnego sprz¦tu i oprogramowania