

METODE DE PLANIFICARE A PROCESELOR ÎN SISTEMELE DE TIMP REAL

Scopul lucrării :

- Descrierea unor metode de planificare a proceselor
- Exemplificarea acestor metode cu ajutorul programului QNET

1. Descrierea generală a programului QNET. Utilizarea acestuia pentru studiul metodelor de planificare a proceselor

1.1. Definirea proceselor și a planificatorului în QNET.

Programul QNET (Queuing Network Tool) reprezintă o interfață grafică pentru un program de analiză a rețelelor cu cozi de așteptare QNAP (Queuing Network Analysis Package). Fiecare *obiect* (*stație sau server*) din structura rețelei cu cozi de așteptare poate fi interpretat în termenii unui sistem de timp real astfel : fiecare *stație* este asimilată cu un proces sau cu o instanță de proces, iar stația *server* este asimilată cu planificatorul de procese.

Procesele pot cere cuante de timp pentru execuție după o lege determinată, iar planificatorul pune în execuție procesele după anumite reguli. Procesele pot fi organizate cu priorități fixe.

Programul QNET permite , din punctul de vedere al studiului metodelor de planificare a proceselor secvențiale, crearea obiectelor de lucru (proces și planificator) definirea caracteristicilor acestora (proprietățile proceselor și regulile ce guvernează planificatorul) și simularea execuției proceselor secvențiale (alocarea cuantelor de timp pentru execuție).

Etapele necesare pentru utilizarea programului QNET în vederea studiului planificării proceselor sînt următoarele:

- crearea unui model QNET
- editarea grafică, în cadrul modelului, a obiectelor necesare
- specificarea proprietăților fiecărui obiect (prin selectarea dintr-un *menu* grafic)
- simularea execuției proceselor
- vizualizarea rezultatelor

Inițial se crează un subdirector ce va fi considerat de sistemul grafic de interpretare a comenzilor ca un obiect de tip **DIR** ce va permite accesul la toate operațiile ulterioare și stocarea celorlalte subdirectoare create în decursul operării.

Crearea acestui obiect se face cu comanda (\$ este prompt-ul UNIX):

```
$ make dir NAME=nume_model_qnet
```

Se crează un subdirector **nume_model_qnet.dir**; acest subdirector devine directorul curent de lucru cu comanda :

```
$ cd nume_model_qnet.dir
```

Lansarea în execuție a programului QNET se efectuează cu comanda în directorul curent astfel :

```
$ xdtm
```

Se lansează în execuție interfața grafică de lucru cu programul QNET. Selectarea comenzilor se va efectua cu ajutorul *mouse-ului* astfel : butonul din dreapta selectează un *menu*, iar butonul din stînga selectează comanda aleasă.

Crearea unui model QNET (subdirector QNET) se realizează din *menu* grafic al obiectului **Dir** (simbolizat prin ".") selectînd operația *create* și tipul de obiect *qnet* (se va cere și un nume pentru modelul QNET).

Pentru modelul astfel creat se pot selecta următoarele comenzi :

1. *Edit* : permite crearea grafică a descrierii modelului QNET (proces, planificator) și definirea caracteristicilor acestora.
2. *Run* : se simulează execuția proceselor conform caracteristicilor acestora și a regulii de planificare.

2. Descrierea editorului grafic QNET

Editorul grafic QNET permite specificarea unui model ce conține procese și planificatorul de procese ale căror caracteristici pot fi modificate ușor prin intermediul ferestrelor grafice cu opțiuni pentru atributele fiecărui nod în parte.

Etapele de lucru sînt descrise în continuare :

2.1. Creare obiecte : se activează, cu butonul din stînga al *mouse-ului*, setul de comenzi și se selectează, cu butonul din dreapta al *mouse-ului*, comanda *Create node*.

Se vor utiliza obiectele : *Source* - pentru procese, *Server Station* - pentru planificator și *Sink* - pentru a indica execuția proceselor .

Aceste obiecte sînt descrise grafic ca în figura 1.

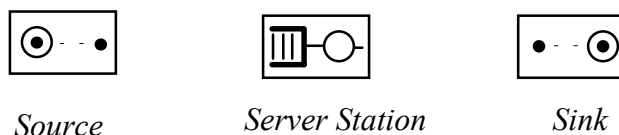


Figura 1

Pentru fiecare proces existent se va crea un obiect *Source*, pentru planificator se va crea un singur obiect *Server Station* și un singur obiect *Sink*.

2.2, Legarea obiectelor : în funcție de obiect se vor utiliza următoarele tipuri de legături (*link*) - de la *Source* la *Server Station* legătura *New-Flow*, iar de la *Server Station* la *Sink* legătura *Flow*.

Se activează, pentru fiecare obiect în parte, cu butonul din stînga al *mouse-ului*, setul de comenzi și se selectează, cu butonul din dreapta al *mouse-ului*, comanda *Make Link*.

În figura 2 este prezentată o posibilă configurație cu 3 procese :

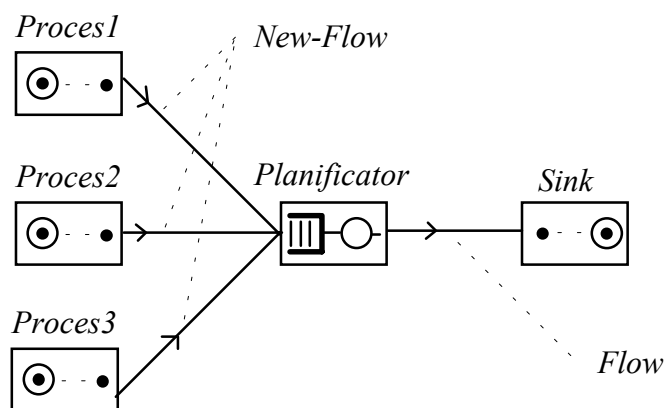


Figura 2

2.3. Definirea atributelor obiectelor : se activează, pentru fiecare obiect în parte, cu butonul din stînga al *mouse-ului*, setul de comenzi și se selectează, cu butonul din dreapta al *mouse-ului*, comanda *Set node attributes*.

Se specifică, în fereastra grafică, următoarele caracteristici ale nodului :

- pentru proces (nod *Source*) : Numele (în câmpul *Station*) și intervalul de apariție a cererilor de execuție (în câmpul *Value*) .

Modul de servire va fi specificat de utilizator astfel : câmpul *Service* subcâmpul *User-code* și se vor introduce următoarele instrucțiuni - pentru fiecare proces în parte (& reprezintă comentarii) :

```
&service process1
BEGIN
& Insert your code here...
PRIOR(0);           &prioritatea
CST(10);            &intervalul între cererile de execuție
& afiseaza momentul de timp la care s-a servit executia si instanta de proces
PRINT(TIME," : ","PROCES1(",TIME,"));
END;
```

```
&service process2
BEGIN
& Insert your code here...
PRIOR(1);           &prioritatea
CST(20);            &intervalul între cererile de execuție
& afiseaza momentul de timp la care s-a servit executia si instanta de proces
PRINT(TIME," : ","PROCES2(",TIME,"));
END;
```

.....

Prioritatea minimă este 0 .

-pentru planificator (nod *Server Station*) : Numele (în câmpul *Server Name*), metoda de planificare (în câmpul *Sched*). Se va alege una din metodele de planificare descrise în secțiunea 3).

Se va completa câmpul *Class* (se vor specifica toate clasele definite ca atribute ale *link-urilor*- vezi punctul 4).

SISTEME DE TIMP REAL PENTRU TELECOMUNICAȚII

LUCRAREA DE LABORATOR NR. 4

Cîmpul *Services per class*, subcîmpul *Distribution* va fi completat astfel : se va selecta cîmpul *User-code* și se vor introduce următoarele instrucțiuni (& reprezintă comentariu):

```
BEGIN
& Insert your code here...
CST(15);
PRINT(TIME,"",CCLASS,"(",CUSTOMER.ENTERDATE,"");
PRINT("____");
PRINT("____");
END;
```

- nodul *Sink* nu are atribute.

2.4. Definirea atributelor pentru legături : Se definesc *clasele* proceselor. Prin definirea claselor se deosebesc cererile de execuție între ele.

Se activează, pentru fiecare *link* în parte, cu butonul din stînga al *mouse-ului*, setul de comenzi și se selectează, cu butonul din dreapta al *mouse-ului*, comanda *Set link attributes*.

Se specifică, în fereastra grafică, următoarele caracteristici ale *link-ului* :

Multiclass, *Name* și *Prob* = 1.

Legătura către *Sink* nu are atribute.

2.5. Definirea atributelor modelului : se activează, cu butonul din stînga al *mouse-ului*, setul de comenzi și se selectează, cu butonul din dreapta al *mouse-ului*, comanda *Set model attributes*.

Se completează următoarele cîmpuri :

Compilation flags :

Source display during compilation : No

Reports Flags:

Standard results on screen : No

Classes :

se scriu toate clasele definite anterior

Solution method : *simulation*

Pentru cîmpul *Simulation Control* se vor completa următoarele subcîmpuri :

Animation Trace : no

Lenght : (intervalul de timp de simulare)

Observații :

- Cîmpurile nespecificate anterior nu se vor modifica
- După orice modificare se va salva fereastra grafică cu comanda *Save*. Ieșirea din fereastra grafică se realizează cu comanda *Done*.

2.6. Generarea codului pentru simulare : se activează, cu butonul din stînga al *mouse-ului*, setul de comenzi și se selectează, cu butonul din dreapta al *mouse-ului*, comanda *Generate code & specs*.

După parcurgerea etapelor 2.1-2.6 se poate simula execuția proceselor, cu comanda *Run* selectată pentru modelul astfel definit. Se activează, cu butonul din stînga al *mouse-ului*, pentru modelul QNET, setul de comenzi și se selectează, cu butonul din dreapta al *mouse-ului*, comanda *Run*.

Rezultatele simulării sînt scrise în fișierul : **single.plan/1.run/qnap2.log** din subdirectorul asociat modelului QNET.

3. Metode de planificare a proceselor

Printre metodele de planificare a proceselor secvențiale se pot enumera :

3.1. Metoda FIFO (First In First Out)

Procese (instanțele de proces) sînt plasate într-o coadă (virtuală) în ordinea sosirii - a cererii execuției .

Primul proces care cere execuția este servit (este pus în execuție).

3.2. Metoda LIFO (Last In First Out)

Procese (instanțele de proces) sînt plasate într-o stivă (virtuală) în ordinea sosirii - a cererii execuției .

Ultimul proces care cere execuția este servit (este pus în execuție).

3.3. Metoda FIFO cu priorități

Procese (instanțele de proces) sînt plasate într-o coadă (virtuală) în ordinea sosirii - a cererii execuției .

Procese de prioritate maximă care cer execuția vor fi servite mai întîi; pentru procese de prioritate egală se aplică regula FIFO.

3.4. Metoda LIFO cu priorități

Procese (instanțele de proces) sînt plasate într-o stivă (virtuală) în ordinea sosirii - a cererii execuției .

Procese de prioritate maximă care cer execuția vor fi servite mai întîi; pentru procese de prioritate egală se aplică regula LIFO.

3.5. Metoda FIFO cu priorități, PREEMPT

Procese (instanțele de proces) sînt plasate într-o coadă (virtuală) în ordinea sosirii - a cererii execuției .

Procese de prioritate maximă care cer execuția vor fi servite mai întîi; pentru procese de prioritate egală se aplică regula FIFO.

În mod suplimentar metodei 3.3 procesele prioritare pot întrerupe procesele de prioritate inferioară care sînt trecute în coada de așteptare FIFO.

3.6. Metoda LIFO cu priorități, PREEMPT

Procese (instanțele de proces) sînt plasate într-o stivă (virtuală) în ordinea sosirii - a cererii execuției .

Procese de prioritate maximă care cer execuția vor fi servite mai întîi; pentru procese de prioritate egală se aplică regula LIFO.

SISTEME DE TIMP REAL PENTRU TELECOMUNICAȚII
LUCRAREA DE LABORATOR NR. 4

În mod suplimentar metodei 3.4 procesele prioritare pot întrerupe procesele de prioritate inferioară care sînt trecute în stiva LIFO.

4. Exemple de planificare

& REZULTATE PLANIFICARE FIFO (fara priortate)

& TIMPUL DE SERVIRE AL PLANIFICATORULUI = 15 cuante

Starting execution of Qnet...

10.00	:	PROCES1(10.00)	
20.00	:	PROCES2(20.00)	
20.00	:	PROCES1(20.00)	
25.00					PROCES1 (10.00)

30.00	:	PROCES1(30.00)	
40.00	:	PROCES2(40.00)	
40.00					PROCES2 (20.00)

40.00	:	PROCES1(40.00)	
50.00	:	PROCES1(50.00)	
55.00					PROCES1 (20.00)

60.00	:	PROCES2(60.00)	
60.00	:	PROCES1(60.00)	
70.00					PROCES1 (30.00)

70.00	:	PROCES1(70.00)	
80.00	:	PROCES2(80.00)	
80.00	:	PROCES1(80.00)	
85.00					PROCES2 (40.00)

90.00	:	PROCES1(90.00)	
100.0	:	PROCES2(100.0)	
100.0					PROCES1 (40.00)

100.0	:	PROCES1(100.0)	
-------	---	----------	-------	---	--

Dumping simulation results...

Results written.

End of Qnet execution.

& REZULTATE PLANIFICARE FIFO (cu priortate)

& TIMPUL DE SERVIRE AL PLANIFICATORULUI = 15 cuante

Starting execution of Qnet...

10.00	:	PROCES1(10.00)	
20.00	:	PROCES2(20.00)	
20.00	:	PROCES1(20.00)	

METODE DE PLANIFICARE A PROCESELOR ÎN SISTEMELE DE TIMP REAL

25.00 PROCES1 (10.00)

30.00 : PROCES1(30.00)
 40.00 : PROCES2(40.00)
 40.00 PROCES2 (20.00)

40.00 : PROCES1(40.00)
 50.00 : PROCES1(50.00)
 55.00 PROCES2 (40.00)

60.00 : PROCES2(60.00)
 60.00 : PROCES1(60.00)
 70.00 PROCES1 (20.00)

70.00 : PROCES1(70.00)
 80.00 : PROCES2(80.00)
 80.00 : PROCES1(80.00)
 85.00 PROCES2 (60.00)

90.00 : PROCES1(90.00)
 100.0 : PROCES2(100.0)
 100.0 PROCES2 (80.00)

100.0 : PROCES1(100.0)
 Dumping simulation results...
 Results written.
 End of Qnet execution.

& REZULTATE PLANIFICARE FIFO (cu prioritate - preempt)
 & TIMPUL DE SERVIRE AL PLANIFICATORULUI = 15 cuante

Starting execution of Qnet...

10.00 : PROCES1(10.00)
 20.00 : PROCES2(20.00)
 20.00 : PROCES1(20.00)
 30.00 : PROCES1(30.00)
 35.00 PROCES2 (20.00)

40.00 : PROCES2(40.00)
 40.00 : PROCES1(40.00)
 50.00 : PROCES1(50.00)
 55.00 PROCES2 (40.00)

SISTEME DE TIMP REAL PENTRU TELECOMUNICAȚII
LUCRAREA DE LABORATOR NR. 4

55.00 PROCES1 (10.00)

60.00 : PROCES2(60.00)
60.00 : PROCES1(60.00)
70.00 : PROCES1(70.00)
75.00 PROCES2 (60.00)

80.00 : PROCES2(80.00)
80.00 : PROCES1(80.00)
90.00 : PROCES1(90.00)
95.00 PROCES2 (80.00)

100.0 : PROCES2(100.0)
100.0 : PROCES1(100.0)

Dumping simulation results...

Results written.

End of Qnet execution.

& REZULTATE PLANIFICARE LIFO (fara prioritate)

& TIMPUL DE SERVIRE AL PLANIFICATORULUI = 15 cuante

Starting execution of Qnet...

10.00 : PROCES1(10.00)
20.00 : PROCES2(20.00)
20.00 : PROCES1(20.00)
25.00 PROCES1 (10.00)

30.00 : PROCES1(30.00)
40.00 : PROCES2(40.00)
40.00 PROCES1 (20.00)

40.00 : PROCES1(40.00)
50.00 : PROCES1(50.00)
55.00 PROCES2 (40.00)

60.00 : PROCES2(60.00)
60.00 : PROCES1(60.00)
70.00 PROCES1 (50.00)

70.00 : PROCES1(70.00)
80.00 : PROCES2(80.00)
80.00 : PROCES1(80.00)
85.00 PROCES1 (60.00)

METODE DE PLANIFICARE A PROCESELOR ÎN SISTEMELE DE TIMP REAL

```

90.00 : PROCES1( 90.00 )
100.0 : PROCES2( 100.0 )
100.0                                PROCES1 ( 80.00 )

```

```

100.0 : PROCES1( 100.0 )
Dumping simulation results...
Results written.
End of Qnet execution.

```

& REZULTATE PLANIFICARE LIFO (cu prioritate)

& TIMPUL DE SERVIRE AL PLANIFICATORULUI = 15 cuante

Starting execution of Qnet...

```

10.00 : PROCES1( 10.00 )
20.00 : PROCES2( 20.00 )
20.00 : PROCES1( 20.00 )
25.00                                PROCES1 ( 10.00 )

```

```

30.00 : PROCES1( 30.00 )
40.00 : PROCES2( 40.00 )
40.00                                PROCES2 ( 20.00 )

```

```

40.00 : PROCES1( 40.00 )
50.00 : PROCES1( 50.00 )
55.00                                PROCES2 ( 40.00 )

```

```

60.00 : PROCES2( 60.00 )
60.00 : PROCES1( 60.00 )
70.00                                PROCES1 ( 50.00 )

```

```

70.00 : PROCES1( 70.00 )
80.00 : PROCES2( 80.00 )
80.00 : PROCES1( 80.00 )
85.00                                PROCES2 ( 60.00 )

```

```

90.00 : PROCES1( 90.00 )
100.0 : PROCES2( 100.0 )
100.0                                PROCES2 ( 80.00 )

```

```

100.0 : PROCES1( 100.0 )
Dumping simulation results...
Results written.
End of Qnet execution.

```

& REZULTATE PLANIFICARE LIFO (cu prioritate - preempt)
& TIMPUL DE SERVIRE AL PLANIFICATORULUI = 15 cuante

Starting execution of Qnet...

10.00	:	PROCES1(10.00)
20.00	:	PROCES2(20.00)
20.00	:	PROCES1(20.00)
30.00	:	PROCES1(30.00)
35.00			PROCES2	(20.00)

40.00	:	PROCES2(40.00)
40.00	:	PROCES1(40.00)
50.00	:	PROCES1(50.00)
55.00			PROCES2	(40.00)

60.00	:	PROCES2(60.00)
60.00	:	PROCES1(60.00)
70.00	:	PROCES1(70.00)
75.00			PROCES2	(60.00)

80.00	:	PROCES2(80.00)
80.00	:	PROCES1(80.00)
90.00	:	PROCES1(90.00)
95.00			PROCES2	(80.00)

100.0	:	PROCES2(100.0)
100.0	:	PROCES1(100.0)

Dumping simulation results...

Results written.

End of Qnet execution.

4. Desfășurarea lucrării

Pentru lansarea în execuție a programului **xdtm** sînt necesare următoarele comenzi (în sistemul de operare UNIX, pe o mașină locală identificată prin *id_mașina_locală*) :

- **startx** ---> se intră în modul de lucru cu ferestre grafice
- din mediul de lucru grafic se deschide o consolă XTERM; comenzile următoare se vor da din XTERM.

- **xhost** *id_mașina_XDTM* ---> se indică faptul că se vor importa ferestre grafice de pe mașina Unix pe care este instalat programul XDTM identificată prin

id_mașina_XDTM

- **telnet** *id_mașina_XDTM*

Următoarele comenzi se vor da de pe mașina distantă *id_mașina_XDTM* :

- **export DISPLAY =** *id_mașina_locală* : 0.0 ---> se indică unde se vor transfera ferestrele grafice

- **cd** *work.dir* ---> se intră în subdirectorul de lucru identificat prin *work.dir*

- **xdtm** ---> se lansează pachetul de programe XDTM

a). Se va studia modul de lucru al programului QNET.

b). Se vor crea diferite modele pentru metodele de planificare a proceselor descrise în secțiunea 2.