Operations Research - Queueing

Konorski Equations

Oznaczenia

a_{sr} śreni interwał przybycia zgłoszenia (zawsze dane ?) avg. time of request arrival (always given?)

b_{sr} średni rozmiar zgłoszenia avg. size of request

v $[wymiarzgloszenia/s]$ prędkość procesora [unit-of-request/s] speed of CPU

τ_{sr} średni czas obsługi zgłoszenia avg time of request handling (finish time?)

w_{sr} średni opóźnienie buforowania avg delay of buffering

d_{sr} średni czas przebywania w systemie (czas życia) avg time of staying in system (life time)

r $[bezwymiarowe]$ współczynnik obciążenie procesora | niestabilność przy $r \geq 1$ [unit-less] coef of CPU utilisation | unstability at r >=1

ρ $[bezwymiarowe]$ normatywny współczynnik obciążenia - w systemach wielo procesorowych [unit-less] norm. coef of utilisation in MCPU

L frakcja zgłoszeń utraconych wskutek przepełnienia pamięci buforowej fraction of requests lost due to buffer overflow

Oznaczenia

a_{sr} śreni interwał przybycia zgłoszenia (zawsze dane ?) avg. time of request arrival (always given?)

b_{sr} średni rozmiar zgłoszenia avg. size of request

v $[wymiarzgloszenia/s]$ prędkość procesora [unit-of-request/s] speed of CPU

τ_{sr} średni czas obsługi zgłoszenia avg time of request handling (finish time?)

w_{sr} średni opóźnienie buforowania avg delay of buffering

d_{sr} średni czas przebywania w systemie (czas życia) avg time of staying in system (life time)

r $[bezwymiarowe]$ współczynnik obciążenie procesora | niestabilność przy $r \geq 1$ [unit-less] coef of CPU utilisation | unstability at r >=1

ρ $[bezwymiarowe]$ normatywny współczynnik obciążenia - w systemach wielo procesorowych [unit-less] norm. coef of utilisation in MCPU

L frakcja zgłoszeń utraconych wskutek przepełnienia pamięci buforowej fraction of requests lost due to buffer overflow

p_k $[bezwymiarowe]$ frakcja czasu w której w systemie jest k zgłoszeń [unit-less] fraction of time at which system has k requests

$1 - p_0$ $[bezwymiarowe]$ współczynnik zajętości (wykorzystania) procesora [unit-less] coef of CPU occupancy/(un)utilisation

$N_{sr}[bezwymiarowe]$ średnia ilość zadań w systemie [unitless] - avg no of tasks in system

Wzory

$\tau_{sr} = \frac{b_{sr}}{v}$ avg time of request handling (finish time?)

$r = \frac{b_{sr}}{a_{sr} * v}$ [unit-less] coef of CPU utilisation | unstability at r >=1

$r = \frac{\tau_{sr}}{a_{sr}}$

Prawo Little'a: Little's Rule:

$N_{sr} = \frac{1-L}{a_{sr}} * d_{sr}$ [unitless] - avg no of tasks in system

w ujęciu procesora: $1 - p_0 = \frac{(1-L)}{a_{sr}} * \tau_{sr} = (1-L) * r$

1 - [unit-less] coef of CPU occupancy/(un)utilisation =

= (1 - fraction of requests lost due to buffer overflow) / (avg. time of request arrival) * (avg time of request hand.)

= (1 - fraction of requests lost due to buffer overflow)*([unit-less] coef of CPU utilisation)

w ujęciu pamięci buforowej: $N_{sr} - (1 - p_0) = \frac{1-L}{a_{sr}} * w_{sr}$

([unitless] - avg no of tasks in system) - ([unit-less] coef of CPU occupancy/(un)utilisation) =

= (1 - fraction of requests lost due to buffer overflow) / (avg. time of request arrival) * (avg delay of buffering)

Notacja KENDALLA

KENDALL Notation

A / B / S / Q / J

OZNACZENIA: WHERE:

- A - rozkład interwałów zgłoszeń A - distribution of requests intervals
- B - rozkład wielkości wymagań zgłoszeń. B - distribution of requests demanded sizes
- S - liczba procesorów S - no of CPU
- Q - pojemność pamięci buforowej (jeśli nie podany to zakładamy ∞) Q - size of buffer memory (if not given, assume its infinity)
- J - rozmiar populacji źródeł zgłoszeń (opcjonalny parametr) J - size of requests sources population (optional)

Typy rozkładów: Types of distributions

- M - wykładniczy M - expotential
- D - Deterministyczny D - deterministic
- G - Ogólny G - general

Systemy Typu: M/?/?/? Systems of type : M/?/?/?

- możemy rozdzielać stumień zgłoszeń w sposób losowy na inne rzadsze: we may separate stream of requests on random others less dense
- możemy agregować stumień zgłoszeń: we may aggregate stream of requests

Systemy Typu: M/M/?/? - Markowowskie: Systems of type: M/M/?/? - Markowow

Systemy Typu: M/M/1/? Systems of type: M/M/1/?

$N_{sr} = \frac{r}{1-r}$ - co w przypadku gdy r = 1 ? - what in case of r = 1?

$p_k = p_0 * r^k$

$p_0 + p_1 + p_2 + p_3 + ... = 1$

$p_0 = 1 - (p_1 + p_2 + p_3 + ...) \Rightarrow p_0 = 1 - r$

Systemy Typu: M/M/1/Q Systems of type: M/M/1/Q

$p_0 = \frac{1-r}{1-r^{Q+1}}$

$L = \frac{1-r}{1-r^{Q+1}} * r^Q$, gdy $r \neq 1$ when r not 1

$L = \frac{1}{Q+1}$, gdy $r = 1$ when r is 1

$N_{sr} = \sum_{k=0}^{Q} p_k * k$

Jeśli p_k jest stałe to: $N_{sr} = p_k * \sum_{k=0}^{Q} k$ if p_k is constant then: ...

Suma skończonego ciągu arytmetycznego: $\frac{a_1 + a_n}{2} * n$ finite arithmetic sequence sum: ...

Systemy Typu: M/M/S Systems of type: M/M/S

$r = \frac{b_{sr}}{a_{sr} * v * S}$ - współczynnik obciążenia - load coefficient

$\rho = \frac{b_{sr}}{a_{sr} * v}$ - normatywny współczynnik obciążenia [erlangi] - normalized load coef [earlang]

Systemy Typu: M/M/∞ Systems of type: M/M/inf

$N_{sr} = \rho$

$p_k = e^{-\rho * \frac{\rho^k}{k!}}$

Exercises

Ex 1.1, old Ex 1

Problem

Problem 1

Plot the queuing processes $N(t)$ = *queue length* and $U(t)$ = *unfinished work* for an arrival stream specified by $(t_n^+) = (0.5\text{ s}, 2\text{ s}, 5\text{ s}, 6.5\text{ s})$ and $(b_n) = (3\text{ s.u.}, 2\text{ s.u.}, 2\text{ s.u.}, 3\text{ s.u.})$, and for two cases:

(a) there are two processors each serving requests at the speed of $v = 1$ s.u./s (no grading, processor-bound and time-sharing service mode are assumed),
(b) there is one processor serving requests at the speed of $v = 2$ s.u./s (time-sharing service mode is assumed).

Compare the two cases from various viewpoints.

Be careful with the slopes of $U(t)$! What are the merits of case (b)? Is case (a) superior from some point of view?

Zadanie 1

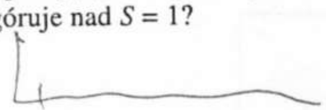
Narysuj przebieg procesów kolejkowych $N(t)$ oraz $U(t)$ = **praca do zakończenia** w systemie masowej obsługi przyjmującego strumień zgłoszeń: $(t_n^+) = (0.5, 2, 5, 6.5)$, $(b_n) = (3, 2, 2, 3)$ dla dwóch przypadków: a) $S = 2$ procesory o wydajności $v = 1$ j.o./s każdy (pełna dostępność, obsługa umiejscowiona), b) $S = 1$ procesor o wydajności $v = 2$ j.o./s. Porównaj te przypadki.

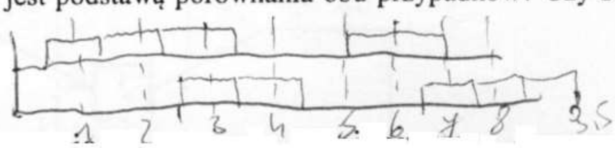
Uwaga na nachylenie opadających części wykresu! Co jest podstawą porównania obu przypadków? Czy $S = 2$ pod jakimś względem góruje nad $S = 1$?

Zadanie 1

Narysuj przebieg procesów kolejkowych $N(t)$ oraz $U(t)$ = **praca do zakończenia** w systemie masowej obsługi przyjmującego strumień zgłoszeń: $(t_n^+) = (0.5, 2, 5, 6.5)$, $(b_n) = (3, 2, 2, 3)$ dla dwóch przypadków: a) $S = 2$ procesory o wydajności $v = 1$ j.o./s każdy (pełna dostępność, obsługa umiejscowiona), b) $S = 1$ procesor o wydajności $v = 2$ j.o./s. Porównaj te przypadki.

Uwaga na nachylenie opadających części wykresu! Co jest podstawą porównania obu przypadków? Czy $S = 2$ pod jakimś względem góruje nad $S = 1$?





Problem short

Narysuj przebieg procesów kolejkowych $N(t)$ oraz $U(t) = pracadozakonczenia$ w systemie masowej obsługi przyjmującego strumień zgłoszeń: (t_n^+) = (0.5, 2, 5, 6.5) $(b_n) = (3, 2, 2, 3)$

dla dwóch przypadków:

a) S = 2 procesory o wydajności v = 1 j.o./s każdy (pełna dostępność, obsługa umiejscowiona),

b) S = 1 procesor o wydajności v = 2 j.o./s. Porównaj te przypadki.

Pytania:

Co jest podstawą porównania obu przypadków?

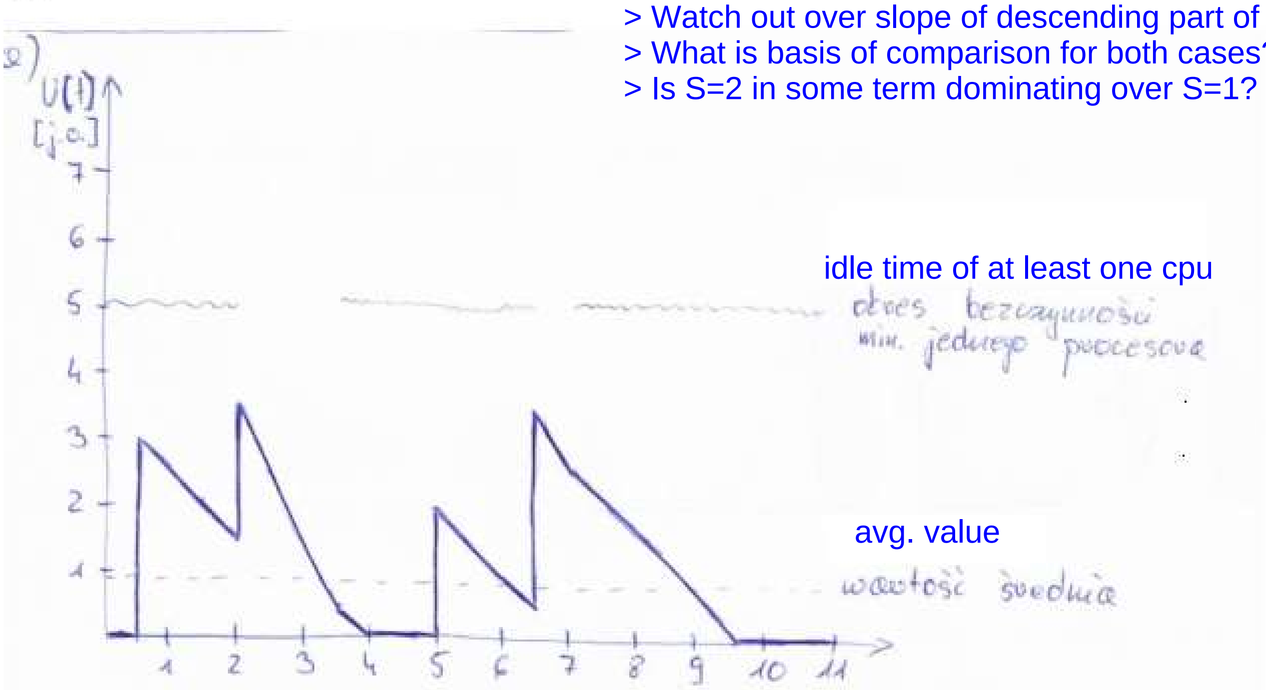
Czy S = 2 pod jakimś względem góruje nad S = 1?

Plot the queueing processes N(t) = queue_length and U(t) = unfinished_work for an arrival stream specified by $(t_n^{+}) = (0.5s, 2s, 5s, 6.5s)$ and $(b_n)=(3 su, 2 su, 2 su, 3su)$ and for two cases:
- a: there are two CPU each serving requests at the speed of v = 1 su/s (no grading, processor-bound and time-sharing service mode are assumed)
- b: there is one processor serving requests at the speed of v = 2 su/s (time-sharing service mode is assumed)

> Compare the two cases from various viewpoints
> Watch out over slope of descending part of plot!
> What is basis of comparison for both cases?
> Is S=2 in some term dominating over S=1?

Solution v1

A)

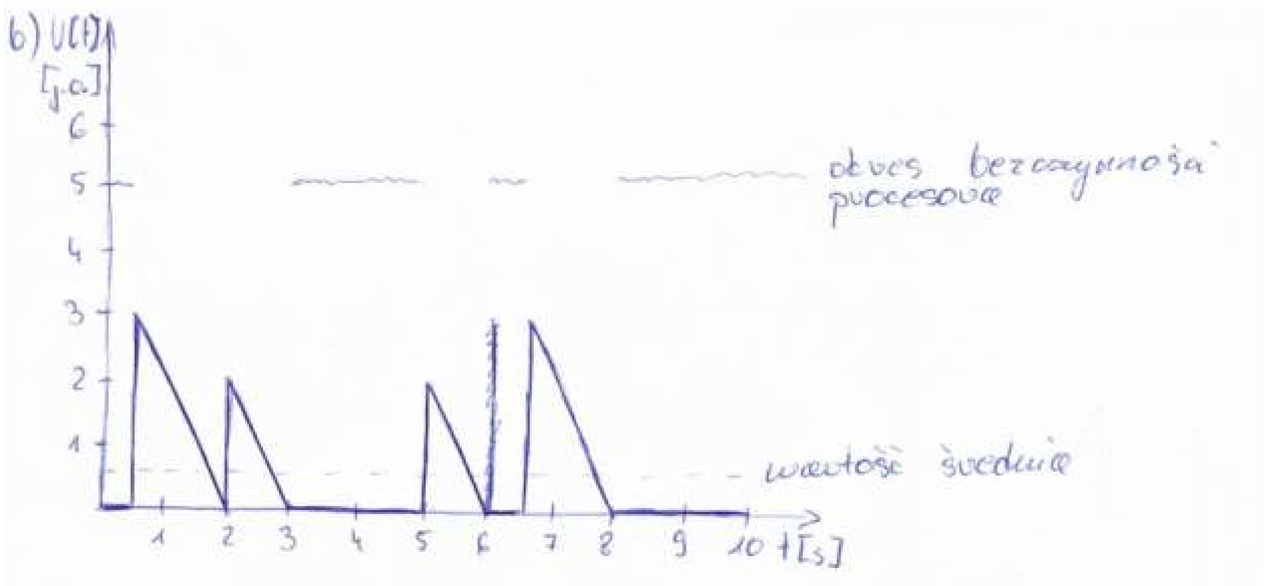


idle time of at least one cpu

avg. value

Schedule:
- 1: at t=0.5 arrives request $b_1=3$
- 2: until t=2 request b_1 is executing (we have left 1.5 of utilisation)
- 3: at t=2 arrives request $b_2=2$
- 4: until t=3.5 we have parallel execution of both requests, where b_1 is finishing and b_2 has 0.5 left to do
- 5: at t=4 we have finish of b_2
- 6: until t=5 cpu's are idle
- 7: at =5 arrives request $b_3=2$
- 8: until t=6.5 executed is b_3 with 0.5 left to do
- 9: at t=6.5 arrives request $b_4=3$
- 10: until t=7 executed are in parallel b_3 and b_4
- 11: at t=7 b_3 finished executing
- 11: until t=9.5 executed is request b_4
- 12: after t=9.5 cpu's are idle

1. w t = 0.5 pojawia się zgłoszenie $b_1 = 3$
2. do t = 2 wykonywane jest zgłoszenie b_1 (pozozstaje 1,5)
3. w t = 2 pojawia się zgłoszenie $b_2 = 2$
4. do t = 3,5 równolegle wykonywane są oba zgłoszenia, gdzie b_1 (się kończy) a b_2 (wynosi jeszcze 0.5)
5. do t = 4 kończy się wykonywanie b_2
6. do t = 5 procesory pozostają bezczynne
7. w t = 5 pojawia się zgłoszenie $b_3 = 2$
8. do t = 6,5 wykonywane jest zgłoszenie b_3 (pozostaje 0.5)
9. w t = 6,5 pojawia się zgłoszenie $b_4 = 3$
10. do t = 7 wykonywane są równolegle zgłoszenia b_3 i b_4
11. do t = 9,5 wykonywane jest zgłoszenie b_4



1. w t = 0.5 pojawia się zgłoszenie $b_1 = 3$
2. zgłoszenie b_1 wykonywane jest do t = 2
3. w t = 2 pojawia się zgłoszenie $b_2 = 2$
4. zgłoszenie b_2 wykonywane jest do t = 3
5. w t = 5 pojawia się zgłoszenie $b_3 = 2$
6. zgłoszenie b_3 wykonywane jest do t = 6
7. w t = 6,5 pojawia się zgłoszenie $b_4 = 3$
8. zgłoszenie b_4 wykonywane jest do t = 8

Schedule:
- 1: at t=0.5 arrives request b_1 = 3
- 2: until t=2 request b_1 is executing
- 3: at finish of b_1 when t=2 appears request b_2 = 2
- 4: until t=3 request b_2 is executing
- 5: in range [3,5] is idle time
- 6: at t=5 appears request b_3 = 2
- 7: until t=6 request b_3 is executing
- 8: in range [6, 6.5] is idle time
- 9: at t=6.5 appears request b_4 = 3
- 10: until t=8 request is executing
- 11: from t=8 we have idle time

Co jest podstawą porównania obu przypadków? podstawą porównania obu przypadków jest wartość średnia. Ans: Avg value

Czy S = 2 pod jakimś względem góruje nad S = 1? Lepiej wykorzystuje procesor (krótszy czas bezczynności)

Ans: S=2 is better utilising cpu (lesser idle time)

Ex 1.2, old Ex 2

Problem

Zadanie 2

Porównaj średnie opóźnienie systemowe zgłoszenia w 1-procesorowym systemie kolejkowym z dyscypliną obsługi FIFO oraz *Round Robin* z **kwantem obsługi** 2 s (niepełne wykorzystanie kwantu obsługi powoduje wcześniejsze rozpoczęcie kolejnego kwantu obsługi). Wydajność procesora wynosi 1 j.o./s. Trzy zgłoszenia: X, Y i Z, o wymaganiach odpowiednio 7 j.o., 1 j.o. i 3 j.o., przybywają jednocześnie i ustawiają się w kolejności a) XYZ, b) YZX, c) XZY.

Czy z porównania opóźnień zgłoszeń dla poszczególnych scenariuszy wynikają ogólne cechy FIFO i RR?

Zadanie 2

Porównaj średnie opóźnienie systemowe zgłoszenia w systemie kolejkowym z dyscypliną obsługi FIFO oraz *Round Robin* z **kwantem obsługi** 2 s (niewykorzystany kwant obsługi przechodzi na następne zadanie). Wydajność procesora wynosi 1 j.o./s. Trzy zgłoszenia: X, Y i Z, o wymaganiach odpowiednio 7 j.o., 1 j.o. i 3 j.o., przybywają jednocześnie i ustawiają się w kolejności a) XYZ, b) YZX, c) XZY.

Czy z porównania opóźnień *poszczególnych* zgłoszeń wynikają ogólne cechy FIFO i RR?

Problem 2

Compare mean system delays in a single-processor queuing system under FIFO and RR with service quantum 2 s (partial use of assigned quantum causes earlier commencement of the next quantum). Processor speed is 1 s.u./s. Three requests, X, Y and Z, of sizes 7 s.u., 1 s.u. and 3 s.u., respectively, arrive simultaneously and queue up in the order (a) XYZ, (b) YZX, (c) XZY.

Can any general properties of FIFO and RR be inferred from such a limited set of scenarios?

Problem short

Porównaj średnie opóźnienie systemowe zgłoszenia w 1-procesorowym systemie kolejkowym.

Z dyscypliną obsługi FIFO oraz Round Robin z kwantem obsługi *2s* (niepełne wykorzystanie kwantu obsługi powoduje wcześniejsze rozpoczęcie kolejnego kwantu obsługi). Wydajność procesora wynosi $1\frac{j.o}{s}$. Trzy zgłoszenia: X, Y i Z, o wymaganiach odpowiednio 7 j.o., 1 j.o. i 3 j.o., przybywają jednocześnie i ustawiają się w kolejności:

- a - XYZ
- b - YZX
- c - XZY

Compare avg system delay of request
in unicpu scheduling.

Under FIFO and Round Robin with service quantum 2s:
- partial use of assigned quantum causes earlier commencement of the next quantum

Processor speed is 1 su/s

Three requests X, Y and Z of sizes (7, 1, 3) [su]
arrive simultaneously and queue up in order
(a) XYZ, (b) YZX, (c) XZY

> Can any general properties of FIFO and RR be inferred from such a limited set of scenarious

Data

> Kwant obsługi - 2s service quantum = 2s
>
> $X = 7$
>
> $Y = 1$
>
> $Z = 3$
>
> FIFO - obsługa zgłoszeń po kolei FIFO - servicing requests next by next
>
> Round Robin - każde zgłoszenie jest obsługiwane w kolejności ale tylko przez pewien czas

RR - each request is serviced next by next but only for some duration of time

Solution v1

$d_{sr} = \frac{d_1+d_2+d_3}{3}$ - trzy zadania

d_avg = (d_1 + d_2 + d_3) / 3 : three tasks

A) (X - Y - Z)

FIFO

1. wykonujemy x w 7s (kończymy zadanie w 7 sekundzie)
2. wykonujemy y w 1s (kończymy zadanie w 8 sekundzie)
3. wykonujemy z w 3s (kończymy zadanie w 11 sekundzie)

FIFO:
1. execute x in 7s (end task at 7th sec.)
2. execute y in 1s (end task at 8th sec.)
3. execute z in 3s (end task at 1

średnie opóźnienie systemowe $= \frac{7+8+11}{3} = 8\frac{2}{3}s$

avg system delay = (7+8+11)/3 = 26/31th sec.)

Round Robin

1. dwa kwanty czasu dla x (pozostaje x=5)
2. jeden kwant czasu dla y (kończymy zadanie y - w 3 sekundzie)
3. dwa kwanty czasy dla z (pozostaje z=1)
4. dwa kwanty czasu dla x (pozostaje x=3)
5. jeden kwant czasu dla z (kończymy zadanie z - w 8 sekundzie)
6. dwa kwanty czasu dla x (pozostaje x=1)
7. jeden kwant czasu dla x (kończymy zadanie x - w 11 sekundzie)

RR:
1: 2 units of time for x (left is x=5)
2: 1 unit of time for y (finish y at t=3)
3: 2 units of time for z (left is z=1)
4: 2 units of time for x(left is x=3)
5: 1 unit of time for z (finish z at t=8)
6: 2 units of time for x (left is x=1)
7: 1 unit of time for x (finish x at t=11)

avg system delay = (3+8+11)/3 = 22/3

średnie opóźnienie systemowe $= \frac{3+8+11}{3} = 7\frac{1}{3}s$

B) (Y - Z - X)

FIFO

1. wykonujemy y w 1s (kończymy zadanie w 1 sekundzie)
2. wykonujemy z w 3s (kończymy zadanie w 4 sekundzie)
3. wykonujemy x w 7s (kończymy zadanie w 11 sekundzie)

FIFO:
1. execute y in 1s (end task at 1th sec.)
2. execute z in 3s (end task at 4th sec.)
3. execute x in 7s (end task at 11th sec.)

średnie opóźnienie systemowe $= \frac{1+4+11}{3} = 5\frac{1}{3}s$

avg system delay = (1+4+11)/3 = 16/3

Round Robin

1. jeden kwant czasu dla y (kończymy zadanie y - w 1 sekundzie)
2. dwa kwanty czasu dla z (pozostaje z=1)
3. dwa kwanty czasu dla x (pozostaje x=5)
4. jeden kwant czasu dla z (kończymy zadanie z - w 6 sekundzie)
5. dwa kwanty czasu dla x (pozostaje x=3)
6. dwa kwanty czasu dla x (pozostaje x=2)
7. jeden kwant czasu dla x (kończymy zadanie x - w 11 sekundzie)

RR:
1: 1 unit of time for y (finish y at t=1)
2: 2 unit of time for z (left is z=1)
3: 2 units of time for x (left is z=5)
4: 1 units of time for z (finish y at t=6)
5: 2 units of time for x (left is z=3)
6: 2 units of time for x (left is x=1)
7: 1 unit of time for x (finish x at t=11)

średnie opóźnienie systemowe $= \frac{1+6+11}{3} = 6s$

avg system delay = (1+6+11)/3 = 18/3

C) (X - Z - Y)

FIFO

1. wykonujemy x w 7s (kończymy zadanie w 7 sekundzie)
2. wykonujemy z w 3s (kończymy zadanie w 10 sekundzie)
3. wykonujemy y w 1s (kończymy zadanie w 11 sekundzie)

średnie opóźnienie systemowe $= \frac{7+10+11}{3} = 9\frac{1}{3}s$

FIFO:
1. execute x in 7s (end task at 7th sec.)
2. execute z in 3s (end task at 10th sec.)
3. execute y in 1s (end task at 11th sec.)

avg system delay = (7+10+11)/3 = 28/3

Round Robin

1. 2 kwanty czasu dla x (pozostaje x = 5)
2. 2 kwanty czasu dla z (pozostaje z = 1)
3. 1 kwant czasu dla y (kończymy zadanie y - w 5 sekundzie)
4. 2 kwanty czasu dla x (pozostaje x = 3)
5. 1 kwanty czasu dla z (kończymy zadanie z - w 8 sekundzie)
6. 2 kwanty czasu dla x (pozostaje x = 1)
7. jeden kwant czasu dla x (kończymy zadanie x - w 11 sekundzie)

średnie opóźnienie systemowe $= \frac{5+8+11}{3} = 8s$

RR:
1: 2 units of time for x (left is x=5)
2: 2 units of time for z (left is z=1)
3: 1 unit of time for y (finish y at t=5)
4: 2 units of time for x (left is x=3)
5: 1 unit of time for z (finish z at t=8)
6: 2 units of time for x (left is x=1)
7: 1 unit of time for x (finish x at t=11)

avg system delay = (11+8+5)/3 = 24/3

FIFO ($8\frac{2}{3}, 5\frac{1}{3}, 9\frac{1}{3}$ **)** FIFO(26,16,28)/3

RR ($7\frac{1}{3}, 6s, 8s$ **)** RR(22,18,24)/3

Round robin jest mniej wrażliwy na przypadek

RR is less sensitive for randomness of tasks execution order due to fairness of time allocation of execution time. Where some tasks will not dominate others and at least let them execute for some unit time quanta.

Ex 1.3, old Ex 7

Problem

Zadanie 3

System masowej obsługi realizuje średnio w ciągu sekundy 800 transakcji, wymagających wykonania średnio 5000 operacji. Każda przybywająca transakcja otrzymuje do swej dyspozycji procesor o średniej wydajności 4 000 000 operacji na sekundę. Oblicz średnią liczbę transakcji w systemie.

Wykorzystaj prawo Little'a.

Zadanie 7

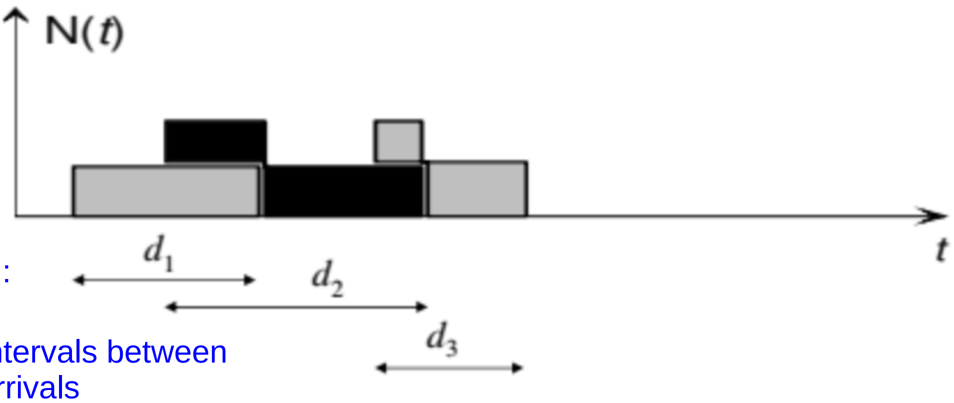
System masowej obsługi realizuje średnio w ciągu sekundy 800 transakcji, wymagających wykonania średnio 5000 operacji. Każda przybywająca transakcja otrzymuje do swej dyspozycji procesor o średniej wydajności 4 000 000 operacji na sekundę. Oblicz średnią liczbę transakcji w systemie.

Wykorzystaj prawo Little'a.

Problem 3

A queuing system serves on average 800 transactions per second, each transaction on average requiring 5000 elementary operations to complete. An arriving transaction is immediately assigned a processor whose speed is 4,000,000 elementary operations/s. Find the mean number of transactions in system.

Use Little's theorem.



tasks delays :

a_m - avg. intervals between next arrivals

T - longer period of time

N_m - mean # requests

$$\text{as } n, T \to \infty, \quad \int_0^T N(t)dt = d_1 + \ldots + d_n$$

$$\boxed{TN_m} \qquad \boxed{(T/a_m)(1-L)d_m}$$

T/a_m - approx. # arrivals

d_m - mean system delay

1 - L : # accepted requests = total # requests - fraction of rejected (overflowed)

approx. # requests = (approx. # arrivals) *(accepted requests) *(mean system delay)

Problem short

cechy zadania: zadanie z prawem little'a task info: exercise with Little Law

System masowej obsługi realizuje średnio w siągu sekundy 800 transakcji, wymagających wykonania średnio 5000 operacji. Każda przybywająca transakcja otrzymuje do swojej dyspozycji procesor o średniej wydajności 4 000 000 operacji na sekundę. Oblicz średnią liczbę transakcji w systemie.

- Queueing system serves 800 transactions/s:
  + where each transaction require on average 5000 elementary operations (eo) to complete.
- Each arriving request is assigned to processor with speed 4M eo/s.
- Find the mean number of transactions in the system.

Solution v1

> Prawo Little'a Średnia populacja = średnia cyrkulacja * średni czas życia

Little's Law: avg population = avg circulation * avg life time

Sposób 1 Method 1

średnia cyrkulacja = $800\frac{transakcji}{s}$ avg circulation = 800 tr/s

średni czas życia = $\frac{5000op}{4000000\frac{op}{s}} = \frac{1}{800}s$ avg life time = 5k[op] / 4M[op/s] = 1/800 [s]

średnia populacja = $800\frac{transakcji}{s} * \frac{1}{800}s = 1 transakcja$ avg population = 800[tr/s]*(1 / 800[s])=1[tr]

Sposób 2 Method 2

$N_{sr} = \frac{(1-L)}{a_{sr}} * d_{sr}, gdzie = d_{sr} = \frac{b_{sr}}{v}$ N_m = (1 - L)/a_m * d_m, where: d_m = b_m / v

> N = liczba zgłoszeń w systemie N_m - avg # requests in system
>
> d - opóźnienie systemowe d_m - avg delay in system
>
> a_{sr} - średni interwał pomiędzy zgłoszeniami a_m - avg interval between requests
>
> b_{sr} - średnie wymaganie zgłoszenia (rozmiar zgłoszenia)[wymagane jednostki operacji] b_m - avg reguirement of requests (size of request) in [reguired op. units]
>
> L - frakcja utraconych transakcji w skutek przepełnienia bufora L - fraction of lost requests due to overflow of buffer
>
> v - wydajność procesora v - performance of cpu

przyjmujemy że bufor jest nieskończony czyli $L = 0$ z treści zadania nie wynika innaczej
assumption: buffer is infinite so L=0 (from exercise context dont seem to be otherwise)

$d_{sr} = \frac{5000op}{4000000\frac{op}{s}} = \frac{1}{800}s$ d_m = 5k[op] / 4M[op/s] = 1 / 800 [s]

$a_{sr} = \frac{1s}{8000op}$ a_m = 1[s] / 8k[op]

$N_{sr} = \frac{1-0}{\frac{1}{800}} * \frac{1}{800} = 1$ N_m = (1 - 0) / (1 / 800) * (1 / 800) = 1

Odpowiedź: Średnia liczba transakcji w systemie wynosi 1. Ans: avg # transaction in system is 1

Ex 2.1, Old ex

Problem

Zadanie 1

Do 1-procesorowego systemu masowej obsługi z procesorem o wydajności $v = 1$ j.o./s przybywa bardzo "gęsty" strumień zgłoszeń, przy czym współczynnik obciążenia wynosi $r = 75\%$. Łączny popyt na obsługę zgłoszony w ciągu sekundy ma odchylenie standardowe $\sigma = 0.1$ s. Jakie są szanse, że w tym okresie uda się wygospodarować pół sekundy pracy procesora dla przetwarzania zadań systemowych bez tworzenia zaległości w obsłudze strumienia zgłoszeń?

Wykorzystaj centralne twierdzenie graniczne i **funkcję Laplace'a**: $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_0^x e^{-y^2/2} dy$.

Jest tablicowana przeważnie dla $x = 0..5$, a jej wybrane wartości wynoszą: $\Phi(0.5) = 0.191$, $\Phi(1) = 0.341$, $\Phi(1.5) = 0.433$, $\Phi(2) = 0.477$, $\Phi(2.5) = 0.4938$, $\Phi(3) = 0.4987$, $\Phi(\geq 4) \approx 0.5$. Co robić, gdy $x < 0$?

Zadanie 1

Do 1-procesorowego systemu masowej obsługi z procesorem o wydajności $v = 1$ j.o./s przybywa bardzo "gęsty" strumień zgłoszeń, przy czym współczynnik obciążenia wynosi $r = 75\%$. Łączny popyt na obsługę zgłoszony w ciągu sekundy ma odchylenie standardowe $\sigma = 0.1$ s. Jakie są szanse, że w tym okresie uda się wygospodarować pół sekundy pracy procesora dla przetwarzania zadań systemowych bez tworzenia zaległości w obsłudze strumienia zgłoszeń?

Wykorzystaj centralne twierdzenie graniczne i **funkcję Laplace'a**: $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_0^x e^{-y^2/2} dy$.

Jest tablicowana przeważnie dla $x = 0..5$, a jej wybrane wartości wynoszą: $\Phi(0.5) = 0.191$, $\Phi(1) = 0.341$, $\Phi(1.5) = 0.433$, $\Phi(2) = 0.477$, $\Phi(2.5) = 0.4938$, $\Phi(3) = 0.4987$, $\Phi(\geq 4) \approx 0.5$. Co robić, gdy $x < 0$?

w każdej 1s wystąpi popyt na 750 ms

Problem 1

gęsty strumień

A single-processor infinite-buffer queuing system with processor speed $v = 1$ s.u./s serves a "dense" arrival stream of requests creating a 75% offered load. The total service demand in a one-second observation period is a random variable with standard deviation $\sigma = 0.1$ s. What are the chances that the processor can spare half of the second to deal with other (e.g., system) tasks without a backlog of requests forming at the end of the observation period?

Use the **Laplace function**: $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_0^x e^{-y^2/2} dy$. Why Laplace function?

Tables of the Laplace function are widely available. Some characteristic values: $\Phi(0.5) = 0.191$, $\Phi(1) = 0.341$, $\Phi(1.5) = 0.433$, $\Phi(2) = 0.477$, $\Phi(2.5) = 0.4938$, $\Phi(3) = 0.4987$, $\Phi(\geq 4) \approx 0.5$. What to do if $x < 0$?

Problem short

Do 1-procesorowego systemu masowej obsługi z procesorem o wydajności $v = 1\frac{j.o}{s}$ przybywa bardzo "gęsty" strumień zgłoszeń przy czym współczynnik obciążenia wynosi $r = 75\%$. Łączny popyt na obsługę zgłoszony w ciągu sekundy ma odchylenie standardowe σ = 0.1 s.

Jakie są szanse, że w tym okresie uda się wygospodarować pół sekundy pracy procesora dla przetwarzania zadań systemowych bez tworzenia zaległości w obsłudze strumienia zgłoszeń?

A unicpu infinite-buffer queueing system with cpu speed v=1[su/s] serves a dense arrival stream of requests creating 75% offered load.

Total service demand in a one-second observation period is a random variable with stdev sigma=0.1[s].

What are the chances that the cpu can spare half of the second to deal with other (eg system) tasks without a backlog of requests forming at the end of the observation period?

Hints

Use central limit theorem

> Wykorzystaj centralne twierdzenie graniczne

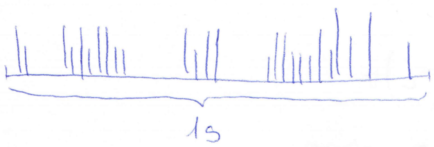
Use Laplace function

> Wykorzystaj funkcję Laplace'a

Laplace function:
Phi(x) = 1/sqrt(2 pi) * int_0^x { exp(-y^2 / 2) dy}

Solution

Zadanie 1 ćwiczenia 2



tau_i = b_i / v : required time of servicing request i

B = 0.5[s] : limit of time for extra tasks
b

$\tau_n = \frac{b_n}{v}$ - wymagany czas obsługi zgłoszenia n

$B = \sum_{n=1}^{\text{dużo}} \tau_n$ - popyt zgłoszony w czasie 1s (suma wysokości słupków)

x = sum_{n=1}^{infinity} tau_n : demand requested in time 1[s] (sum of bars in diagram)

$\bar{B} = v = 75\%$ z treści zadania

$\sigma_B = 0{,}1\ s$ odchylenie standardowe

vec{x} = r = 75% : load coefficient meaning mean in distribution

sigma_x = 0.1[s] : stdev

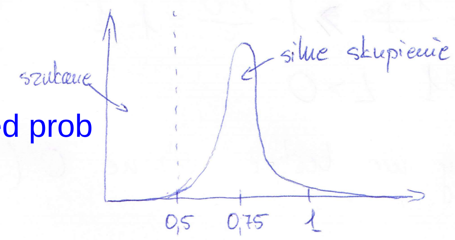
$P(B \leq 0{,}5)$ - szukane prawdopodobieństwo

P(x <= 0.5) : probability searched

Z centralnego twierdzenia granicznego B ma rozkład normalny (Gaussa).

From central bound theorem for n-> infinity, B has normal distribution (Gauss)

searched prob



strong concentration

mu = 0.75 : mean (load coef)

sigma = 0.1 [s] : stdev

$$P(a \leq X_{(v,\sigma)} \leq b) = P(0 \leq X_{(0.75,0.1)} \leq 0.5) =$$

$$= \Phi\left(\frac{b-v}{\sigma}\right) - \Phi\left(\frac{a-v}{\sigma}\right) = \Phi\left(\frac{0.5-0.75}{0.1}\right) - \Phi\left(\frac{0-0.75}{0.1}\right) =$$

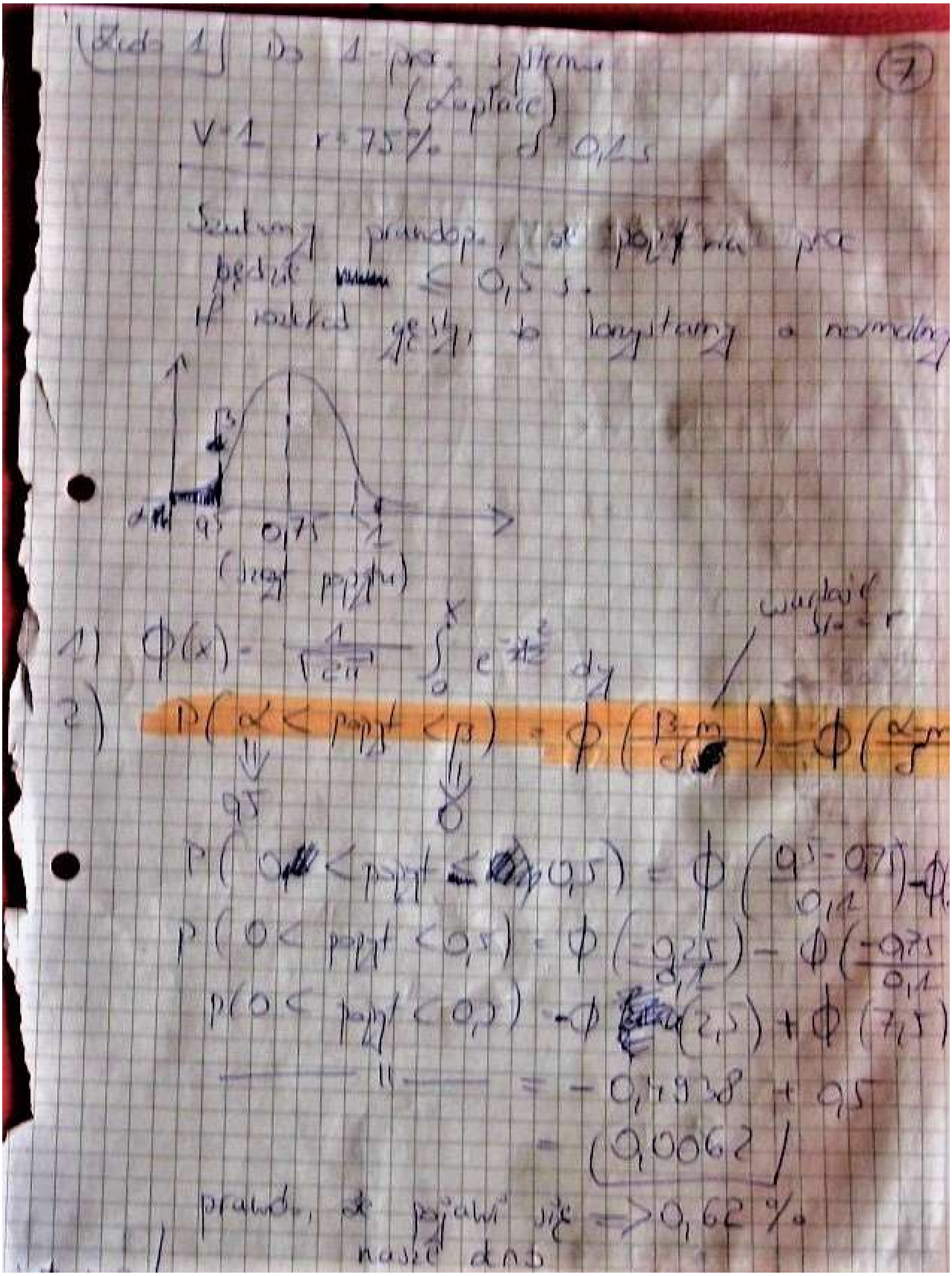
$$= \Phi(-2{,}5) - \Phi(-7{,}5) = \Phi(7{,}5) - \Phi(2{,}5) \approx 0.5 - 0.4938 =$$

$$= 0.0062 = 0{,}62\%$$

Odp. Szanse wynoszą 6,2‰, czyli taka sytuacja wystąpi raz na około 162 sekundy.

Ans. chances are 0.62%, so mentioned situation will happen once every 162[s]

Solution v2



Zad. 1) Do 1-pro. systemu (Laplace)

$V=1$ $r=75\%$ $\sigma=0{,}1s$

Szukamy prawdop., że pojawi się proc. będzie $\le 0{,}5$ s.

W rozkład gęsty, to longitarny a normalny

1) $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_0^x e^{-\frac{t^2}{2}} dy$

wartość śr. = r

2) $P(\alpha < \text{pojaw} < \beta) = \Phi\left(\frac{\beta - m}{\sigma}\right) - \Phi\left(\frac{\alpha - m}{\sigma}\right)$

$P(0 < \text{pojaw} \le 0{,}5) = \Phi\left(\frac{0{,}5 - 0{,}75}{0{,}1}\right) - \Phi(\dots$

$P(0 < \text{pojaw} < 0{,}5) = \Phi\left(\frac{-0{,}25}{0{,}1}\right) - \Phi\left(\frac{-0{,}75}{0{,}1}\right)$

$p(0 < \text{pojaw} < 0{,}5) = -\Phi(2{,}5) + \Phi(7{,}5)$

$= -0{,}4938 + 0{,}5$

$= 0{,}0062$

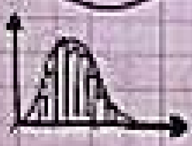
prawd., że pojawi się $\Rightarrow 0{,}62\%$ nasze dno

Solution v3

3) $x \cdot 75\%$ [opóźnienia transakcji]

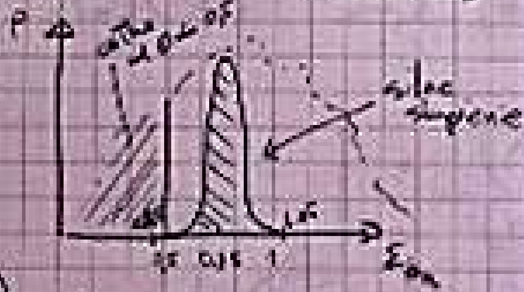
$$\frac{\left(\sum b_n\right)_{śr}}{1s} = x \cdot 75\%$$

2.1



$\sigma = 0{,}1s$ [odchylenie standardowe]

$(\ )_{śr} = 0{,}75s$



a) szansa na 0,5 s 'wolnego'

$$\Phi(x) = \int_0^x \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} dy$$

funkcja Laplace'a

$$\Phi(-x) = -\Phi(x)$$

[całkowita nie granicowa]

$$P(a \le X_{N(m,\sigma)} \le b) = \Phi\left(\frac{b-m}{\sigma}\right) - \Phi\left(\frac{a-m}{\sigma}\right) =$$

$a = 0$, $b = 0{,}5$, $m = 0{,}75$, $\sigma = 0{,}1$

$$= \Phi\left(\frac{0{,}5-0{,}75}{0{,}1}\right) - \Phi\left(\frac{0-0{,}75}{0{,}1}\right) = \Phi(-2{,}5) - \Phi(-7{,}5) =$$

$$= \Phi(7{,}5) - \Phi(2{,}5) = 0{,}62\%$$

$\Phi(7{,}5) = 0{,}5$, $\Phi(2{,}5) = 0{,}4938$

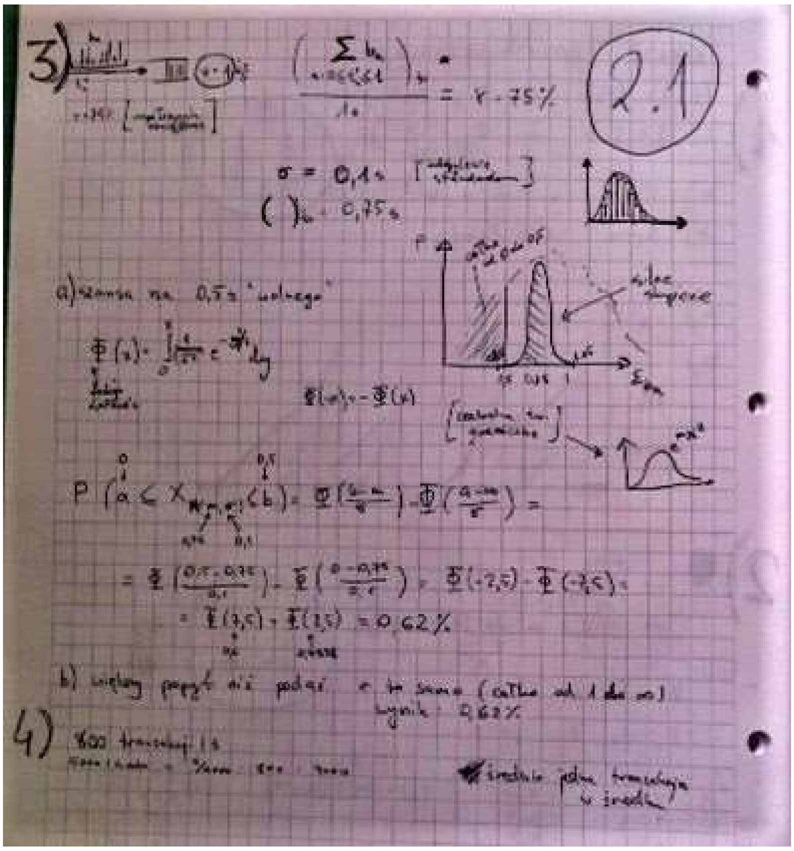
b) większy popyt niż podaż = to samo (całka od 1 do ∞)
wynik: 0,62%

4) 800 transakcji / s

5000 / 1 s ... = ... : 800 : 5000

średnio jedna transakcja w środku

Solution v3 - copy



Ex 2.2, Old ex 3

Problem

Problem 2

A single-processor queuing system with a finite buffer of capacity Q works under offered load $r > 1$. In such a system, the loss fraction L never drops below a certain level. What level is this?

Recall the flow conservation equation. How often is the processor idle if $Q \to \infty$?

ciągłość przepływu

250 ms  100 ms

Mówił że 0,8 s średnio
czy to się uda, jeśli ... co ... sekund?

czy i jak często proc bezcz. w
rozkład normalny
znany średni
i odchylenie standardowe

Zadanie 2

1-procesorowy system masowej obsługi posiada pamięć buforową o pojemności $Q < \infty$ zgłoszeń i pracuje ze współczynnikiem obciążenia $r > 1$. W takim systemie pomimo wzrostu Q frakcja utraconych zgłoszeń L nigdy nie spadnie poniżej pewnego progu – jaką ma on wartość?

Wykorzystaj równanie ciągłości przepływu.
Jak często procesor jest bezczynny, gdy $Q \to \infty$?

Zadanie 2

1-procesorowy system masowej obsługi posiada pamięć buforową o pojemności $Q < \infty$ zgłoszeń i pracuje ze współczynnikiem obciążenia $r > 1$. W takim systemie pomimo wzrostu Q frakcja utraconych zgłoszeń L nigdy nie spadnie poniżej pewnego progu – jaką ma on wartość?

Należy przypomnieć sobie o równaniu ciągłości przepływu. Jak często procesor jest bezczynny, gdy $Q \to \infty$?

$$\frac{1-p_0}{r} = (1-L) \quad L = \frac{1-p_0}{r} - 1$$

$$L = 1 - \frac{1-p_0}{r} \to 0$$

$$1 - \frac{1}{r}$$

Zadanie 3

1-procesorowy system masowej obsługi posiada pamięć buforową o pojemności $Q < \infty$ zgłoszeń i pracuje ze współczynnikiem obciążenia $r > 1$. W takim systemie pomimo wzrostu Q frakcja utraconych zgłoszeń L nigdy nie spadnie poniżej pewnego progu – jaką ma on wartość?

Należy przypomnieć sobie o równaniu ciągłości przepływu. Jak często procesor jest bezczynny, gdy $Q \to \infty$?

Problem short

cechy zadania: zadanie z równaniem ciagłości przeplywu

1-procesorowy system masowej obsługi posiada pamięć buforową o pojemności $Q \leq \infty$ zgłoszeń i pracuje ze współczynnikiem obciążenia $r > 1$. W takim systemie pomimo wzrostu Q frakcja utraconych zgłoszeń : nigdy nie spadnie poniżej pewnego progu - jaką ma on wartość ?

Hints

1. Należy przypomnieć sobie równianie ciągłości przepływu.
2. Jak często procesor jest bezczynny gdy $Q \to \infty$

Equations

> równanie ciągłości przepływu: $1 - p_0 = \frac{1-L}{a_{sr}} * \tau_{sr} = (1 - L) * r$
>
> p_0 współczynik bezczynności procesora L współczynnik strat r współczynnik obciążenia a_{sr} τ_{sr}

Solution v1

$1 - p_0 = (1 - L) * r$

$\frac{1-p0}{r} = \frac{(1-L)*r}{r}$

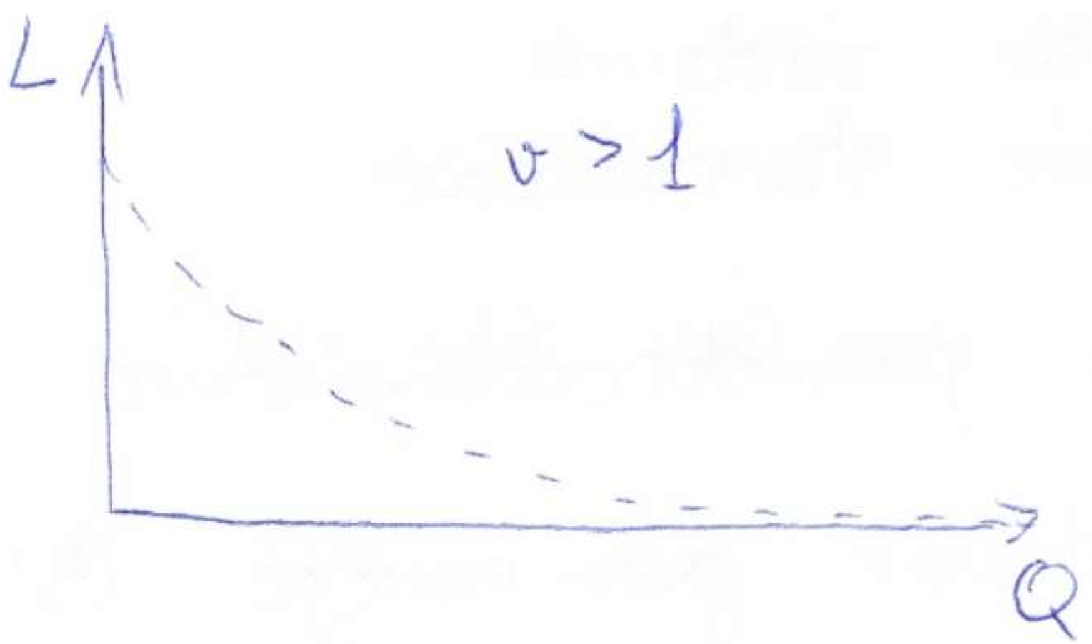
$\frac{1-p0}{r} = (1 - L)$

$\frac{1-p0}{r} - 1 = -L$

$L = 1 - \frac{1-p0}{r}$

wiedząc że $r > 1$ oraz że $Q \to \infty$ można przyjąć że współczynnik bezczynności procesora w takich warunkach wynosi 0 ($p_0 \to 0$)

$L = 1 - \frac{1-p0}{r} \geq 1 - \frac{1-0}{r} = 1 - \frac{1}{r}$ dla $r > 1$



Solution v2

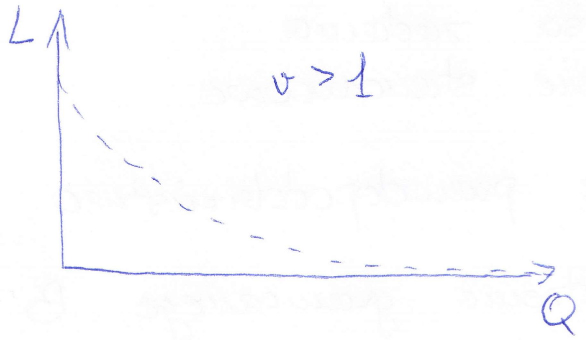
Zadanie 2 ćwiczenie 2

Równanie ciągłości przepływu:

$$1 - p_0 = \frac{1-L}{a_{śr}} \tau_{śr} = (1-L)v$$

p_0 - współczynnik bezczynności procesora

$Q \to \infty$, procesor „się nie wyrabia" ($v>1$), można więc przyjąć, że współczynnik bezczynności procesora wynosi 0 ($p_0 \to 0$).



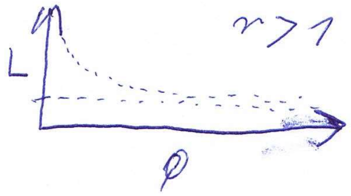
$$L = 1 - \frac{1-p_0}{v} \Rightarrow 1 - \frac{1-0}{v} = 1 - \frac{1}{v}$$

dla $v>1$ $L>0$

Ponieważ $v>1$ L nigdy nie będzie równe 0 (jest to szukany próg).

Solution v3

2/2



$\varrho < \infty$

$r > 1$

$L = ?$

Pytanie o frakcję strat L?

$$\overline{populacja} = \overline{cyrkulacja} \times \overline{czas\ życia}$$

$0 \cdot p_0 + 1(1 - P_0)$

bezczynność

$$1 - P_0 = \frac{1-L}{\varrho m} \times \frac{bm}{V}$$

$$1 - P_0 = (1-L)r$$

$$-P_0 = (1-L)r - 1$$

$$P_0 = (1-L)$$

$$\downarrow \varrho \to \infty$$

$$1 = (1-L)r$$

$$L = 1 - \frac{1}{r},\ r \geq 1$$

~~$P_0 \to \varrho$~~

$P_0 \to 0$

$\varrho \to \infty$

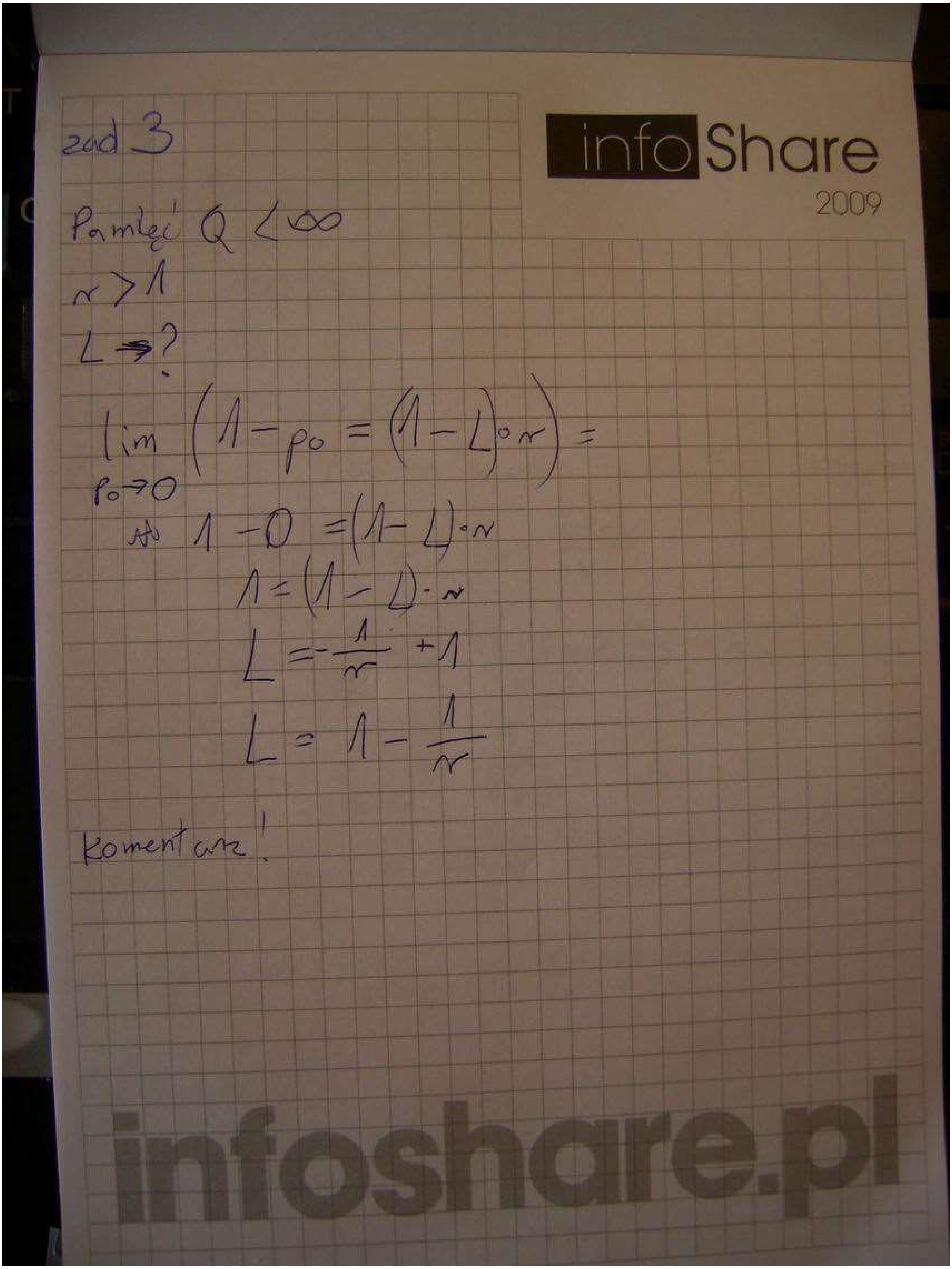
$r = 120\%$

$L_{gr} = 15,7\%$

Niezależnie od ϱ $L = 15,7\%$

System źle zaprojektowany

Solution v4


zad 3
infoShare
2009
Pamięć Q < ∞
r > 1
L → ?
lim (1 − p0 = (1 − L)·r) =
p0→0
1 − 0 = (1 − L)·r
1 = (1 − L)·r
L = −1/r + 1
L = 1 − 1/r
Komentarz!
infoshare.pl

Ex 2.3, Old ex 10

Problem

Problem 3

A processor handles telemetric reports generated by a number of identical sources. To enable multiple access, the processor is equipped with a finite buffer that can accommodate up to Q reports. The table below contains the mean system delay of a report, normalized to the mean report processing time (in boldface), and report loss fraction due to buffer overflow, as dependent on Q and offered load r. Find the maximum number of sources that can be connected to the processor and necessary buffer capacity under the following assumptions:

- each source generates on average 20 reports per minute,
- a report contains on average 1800 records of data,
- the processor is capable of handling 12000 records per second,
- tolerable mean system delay of a report is 1.8 s,
- tolerable loss fraction due to buffer overflow is 4%.

znormalizowane!

r \ Q=	20		21		22		23		24		25	
0.1	**1.11**	0	**1.11**	0	**1.11**	0	**1.11**	0	**1.11**	0	**1.11**	0
0.2	**1.25**	0	**1.25**	0	**1.25**	0	**1.25**	0	**1.25**	0	**1.25**	0
0.3	**1.43**	0	**1.43**	0	**1.43**	0	**1.43**	0	**1.43**	0	**1.43**	0
0.4	**1.67**	0	**1.67**	0	**1.67**	0	**1.67**	0	**1.67**	0	**1.67**	0
0.5	**2**	0	**2**	0	**2**	0	**2**	0	**2**	0	**2**	0
0.6	**2.5**	0	**2.5**	0	**2.5**	0	**2.5**	0	**2.5**	0	**2.5**	0
0.7	**3.32**	0	**3.32**	0	**3.32**	0	**3.33**	0	**3.33**	0	**3.33**	0
0.8	**4.77**	0	**4.8**	0	**4.84**	0	**4.86**	0	**4.89**	0	**4.91**	0
0.9	**7.23**	0.01	**7.42**	0.01	**7.6**	0.01	**7.76**	0.01	**7.92**	0.01	**8.07**	0.01
1	**10.5**	0.05	**11**	0.05	**11.5**	0.04	**12**	0.04	**12.5**	0.04	**13**	0.04
1.1	**13.5**	0.11	**14.3**	0.1	**15.1**	0.1	**15.9**	0.1	**16.7**	0.1	**17.5**	0.1
1.2	**15.5**	0.17	**16.5**	0.17	**17.4**	0.17	**18.4**	0.17	**19.3**	0.17	**20.3**	0.17
1.3	**16.8**	0.23	**17.8**	0.23	**18.7**	0.23	**19.7**	0.23	**20.7**	0.23	**21.7**	0.23
1.4	**17.5**	0.29	**18.5**	0.29	**19.5**	0.29	**20.5**	0.29	**21.5**	0.29	**22.5**	0.29
1.5	**18**	0.33	**19**	0.33	**20**	0.33	**21**	0.33	**22**	0.33	**23**	0.33
1.6	**18.3**	0.38	**19.3**	0.38	**20.3**	0.38	**21.3**	0.38	**22.3**	0.38	**23.3**	0.38
1.7	**18.6**	0.41	**19.6**	0.41	**20.6**	0.41	**21.6**	0.41	**22.6**	0.41	**23.6**	0.41
1.8	**18.8**	0.44	**19.8**	0.44	**20.8**	0.44	**21.8**	0.44	**22.8**	0.44	**23.8**	0.44
1.9	**18.9**	0.47	**19.9**	0.47	**20.9**	0.47	**21.9**	0.47	**22.9**	0.47	**23.9**	0.47
2	**19**	0.5	**20**	0.5	**21**	0.5	**22**	0.5	**23**	0.5	**24**	0.5

maksimum średni dopuszczalny system którym akceptujemy zgłoszeń.

$d \le 12$

$L \le 0,04$

więcej konieczne

Zadanie 3

1-procesorowy system przetwarzania raportów z końcówek telemetrycznych posiada skończoną pamięć buforową o pojemności Q raportów. W zależności od Q i współczynnika obciążenia r zbadano charakterystyki: średniego opóźnienia systemowego raportu, znormalizowanego względem średniego czasu przetwarzania przez procesor (wytłuszczone w tabeli) oraz frakcji raportów utraconych wskutek przepełnienia pamięci buforowej.
Wyznacz maksymalną liczbę jednakowych końcówek telemetrycznych oraz niezbędną pojemność pamięci buforowej przy następujących założeniach:

- końcówka generuje średnio 20 raportów na minutę,
- raport zawiera średnio 1800 rekordów,
- dysponujemy procesorem o wydajności przetwarzania 12000 rekordów na sekundę,
- podział procesora pomiędzy obsługiwane końcówki ma miejsce na zasadzie wspólnego obszaru pamięci buforowej o skończonej pojemności (w raportach),
- dopuszczalne średnie opóźnienie systemowe raportu wynosi 1.8 s,
- dopuszczalna frakcja raportów utraconych wskutek przepełnienia wynosi 4%.

$Q=$	20		21		22		23		24		25	
$r=$												
0.1	**1.11**	0	**1.11**	0	**1.11**	0	**1.11**	0	**1.11**	0	**1.11**	0
0.2	**1.25**	0	**1.25**	0	**1.25**	0	**1.25**	0	**1.25**	0	**1.25**	0
0.3	**1.43**	0	**1.43**	0	**1.43**	0	**1.43**	0	**1.43**	0	**1.43**	0
0.4	**1.67**	0	**1.67**	0	**1.67**	0	**1.67**	0	**1.67**	0	**1.67**	0
0.5	**2**	0	**2**	0	**2**	0	**2**	0	**2**	0	**2**	0
0.6	**2.5**	0	**2.5**	0	**2.5**	0	**2.5**	0	**2.5**	0	**2.5**	0
0.7	**3.32**	0	**3.32**	0	**3.32**	0	**3.33**	0	**3.33**	0	**3.33**	0
0.8	**4.77**	0	**4.8**	0	**4.84**	0	**4.86**	0	**4.89**	0	**4.91**	0
0.9	**7.23**	0.01	**7.42**	0.01	**7.6**	0.01	**7.76**	0.01	**7.92**	0.01	**8.07**	0.01
1	**10.5**	0.05	**11**	0.05	**11.5**	0.04	**12**	0.04	**12.5**	0.04	**13**	0.04
1.1	**13.5**	0.11	**14.3**	0.1	**15.1**	0.1	**15.9**	0.1	**16.7**	0.1	**17.5**	0.1
1.2	**15.5**	0.17	**16.5**	0.17	**17.4**	0.17	**18.4**	0.17	**19.3**	0.17	**20.3**	0.17
1.3	**16.8**	0.23	**17.8**	0.23	**18.7**	0.23	**19.7**	0.23	**20.7**	0.23	**21.7**	0.23
1.4	**17.5**	0.29	**18.5**	0.29	**19.5**	0.29	**20.5**	0.29	**21.5**	0.29	**22.5**	0.29
1.5	**18**	0.33	**19**	0.33	**20**	0.33	**21**	0.33	**22**	0.33	**23**	0.33
1.6	**18.3**	0.38	**19.3**	0.38	**20.3**	0.38	**21.3**	0.38	**22.3**	0.38	**23.3**	0.38
1.7	**18.6**	0.41	**19.6**	0.41	**20.6**	0.41	**21.6**	0.41	**22.6**	0.41	**23.6**	0.41
1.8	**18.8**	0.44	**19.8**	0.44	**20.8**	0.44	**21.8**	0.44	**22.8**	0.44	**23.8**	0.44
1.9	**18.9**	0.47	**19.9**	0.47	**20.9**	0.47	**21.9**	0.47	**22.9**	0.47	**23.9**	0.47
2	**19**	0.5	**20**	0.5	**21**	0.5	**22**	0.5	**23**	0.5	**24**	0.5

Zadanie 10

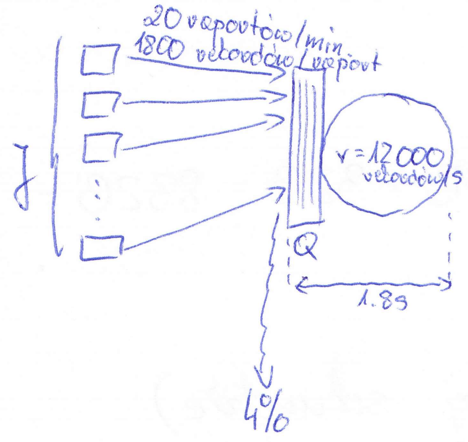
1-procesorowy system przetwarzania raportów z końcówek telemetrycznych posiada skończoną pamięć buforową o pojemności Q raportów. W zależności od Q i współczynnika obciążenia r zbadano charakterystyki: średniego opóźnienia systemowego raportu, znormalizowanego względem średniego czasu przetwarzania przez procesor (wytłuszczone w tabeli) oraz frakcji raportów utraconych wskutek przepełnienia pamięci buforowej.
Wyznacz maksymalną liczbę jednakowych końcówek telemetrycznych oraz niezbędną pojemność pamięci buforowej przy następujących założeniach:

- końcówka generuje średnio 20 raportów na minutę,
- raport zawiera średnio 1800 rekordów,
- dysponujemy procesorem o wydajności przetwarzania 12000 rekordów na sekundę,
- podział procesora pomiędzy obsługiwane końcówki ma miejsce na zasadzie wspólnego obszaru pamięci buforowej o skończonej pojemności (w raportach),
- dopuszczalne średnie opóźnienie systemowe raportu wynosi 1.8 s,
- dopuszczalna frakcja raportów utraconych wskutek przepełnienia wynosi 4%.

$Q=$	20		21		22		23		24		25	
$r=$												
0.1	**1.11**	0	**1.11**	0	**1.11**	0	**1.11**	0	**1.11**	0	**1.11**	0
0.2	**1.25**	0	**1.25**	0	**1.25**	0	**1.25**	0	**1.25**	0	**1.25**	0
0.3	**1.43**	0	**1.43**	0	**1.43**	0	**1.43**	0	**1.43**	0	**1.43**	0
0.4	**1.67**	0	**1.67**	0	**1.67**	0	**1.67**	0	**1.67**	0	**1.67**	0
0.5	**2**	0	**2**	0	**2**	0	**2**	0	**2**	0	**2**	0
0.6	**2.5**	0	**2.5**	0	**2.5**	0	**2.5**	0	**2.5**	0	**2.5**	0
0.7	**3.32**	0	**3.32**	0	**3.32**	0	**3.33**	0	**3.33**	0	**3.33**	0
0.8	**4.77**	0	**4.8**	0	**4.84**	0	**4.86**	0	**4.89**	0	**4.91**	0
0.9	**7.23**	0.01	**7.42**	0.01	**7.6**	0.01	**7.76**	0.01	**7.92**	0.01	**8.07**	0.01
1	**10.5**	0.05	**11**	0.05	**11.5**	0.04	**12**	0.04	**12.5**	0.04	**13**	0.04
1.1	**13.5**	0.11	**14.3**	0.1	**15.1**	0.1	**15.9**	0.1	**16.7**	0.1	**17.5**	0.1
1.2	**15.5**	0.17	**16.5**	0.17	**17.4**	0.17	**18.4**	0.17	**19.3**	0.17	**20.3**	0.17
1.3	**16.8**	0.23	**17.8**	0.23	**18.7**	0.23	**19.7**	0.23	**20.7**	0.23	**21.7**	0.23
1.4	**17.5**	0.29	**18.5**	0.29	**19.5**	0.29	**20.5**	0.29	**21.5**	0.29	**22.5**	0.29
1.5	**18**	0.33	**19**	0.33	**20**	0.33	**21**	0.33	**22**	0.33	**23**	0.33
1.6	**18.3**	0.38	**19.3**	0.38	**20.3**	0.38	**21.3**	0.38	**22.3**	0.38	**23.3**	0.38
1.7	**18.6**	0.41	**19.6**	0.41	**20.6**	0.41	**21.6**	0.41	**22.6**	0.41	**23.6**	0.41
1.8	**18.8**	0.44	**19.8**	0.44	**20.8**	0.44	**21.8**	0.44	**22.8**	0.44	**23.8**	0.44
1.9	**18.9**	0.47	**19.9**	0.47	**20.9**	0.47	**21.9**	0.47	**22.9**	0.47	**23.9**	0.47
2	**19**	0.5	**20**	0.5	**21**	0.5	**22**	0.5	**23**	0.5	**24**	0.5

Solution v1

Zadanie 3 ćwiczenie 2



$$\tau_{3v} = \frac{b_{3v}}{v} = \frac{1800 \frac{\text{rekordów}}{\text{raport}}}{12000 \frac{\text{rekordów}}{s}} = 0{,}15 \ s/\text{raport}$$

$$\frac{1{,}8 s}{0{,}15 \frac{s}{\text{raport}}} = 12 \text{ raportów}$$

~~system stabilny => v < 1~~

frakcja raportów utraconych $\leq 4\%$ => $v \leq 1$

średnie opóźnienie systemowego raportu ≤ 12 i maksymalna liczba końcówek => $Q = 22 \vee Q = 23$

$$v = \frac{b_{3v}}{a_{3v} \cdot V} = \frac{1800 \text{ rekordów}}{3s \cdot 12000 \frac{\text{rekordów}}{s}} = \frac{1800 \text{ rekordów}}{36000 \text{ rekordów}} = 0{,}05 = 5\%$$

raport co 3s

dla jednej końcówki

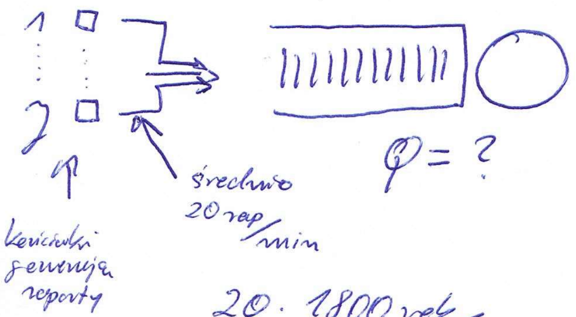
$$J \cdot v \leq 1 \Rightarrow J \leq 20$$

Odp. Maksymalna liczba końcówek to 20, a niezbędna ~~pam~~ pojemność pamięci buforowej wynosi 22.

4

Solution v2

~~3/3 i a właściwie 3/4~~ 2/3



$V = 12000$ rec/s

$20 \cdot 1800$ rek/min

$V = 12000$ rek/s

$a_m = 20$ rap/min $= 3$ sek

~~$d_m \to 1{,}8$ s~~

$L \leq 4\%$

$d_m \leq 1{,}8$ s

$1.11 = \dfrac{d_m}{T_m}$

$r_1 = \dfrac{b_m / a_m}{V} = \dfrac{1}{20}$

r_1 - obciążenie dla 1 końcówki

$r = J \cdot r_1 = J \cdot 5\%$

$r < 1$ ma sens dla nieskończonej kolejki

$T = \dfrac{b_m}{V} = 150$ ms

$d_m = \dfrac{1{,}8}{0{,}15} = 12$

$J = 20 \to \varphi = 22$ lub 23

Ex 3.1, Old ex 4

Problem

nadajnik to procesor

Practice Set 3

Problem 1

Each of 50 terminals connected to a common transceiver generates a request after a think time of average duration $^2/_3$ s. In 80% cases it is a message of average length 1000 bytes, and in 20% cases a control data report of average length 160 bytes. The transceiver works at 1 Mb/s in half-duplex; the average proportion of time it is switched to receive mode is 75% (during that time it is unavailable to the terminals). What is the resulting loss fraction?

Data in the problem permit to identify a_m, b_m and proportion of processor idle time. Now go back to the flow conservation equation...

Zadanie 1

Każda z 50 końcówek sieciowych dołączonych do wspólnego nadajnika pracuje w następujący sposób: faza namysłu trwa średnio $^2/_3$ s, po czym generowane jest zgłoszenie; w 80% przypadków jest to wiadomość (średnio 1000 B), zaś w 20% – raport diagnostyczny (średnio 160 B). Nadajnik pracuje w trybie transmisji półdupleksowej z prędkością 1 Mb/s i w każdej sekundzie średnio przez 750 ms jest przełączony na odbiór (w tym czasie jest niedostępny dla naszych zgłoszeń). Jaka będzie średnia frakcja zgłoszeń utraconych?

Dane pozwalają na identyfikację $a_{śr}$, $b_{śr}$ oraz współczynnika zajętości procesora. Należy teraz powrócić do równania ciągłości przepływu.

Zadanie 4

Każda z 50 końcówek sieciowych dołączonych do wspólnego nadajnika pracuje w następujący sposób: faza namysłu trwa średnio $^2/_3$ s, po czym generowane jest zgłoszenie; w 80% przypadków jest to wiadomość (średnio 1000 B), zaś w 20% – raport diagnostyczny (średnio 160 B). Nadajnik pracuje w trybie transmisji półdupleksowej z prędkością 1 Mb/s i w każdej sekundzie średnio przez 750 ms powinien być przełączony na odbiór (w tym czasie jest niedostępny dla naszych zgłoszeń). Jaka będzie średnia frakcja zgłoszeń utraconych?

Dane pozwalają na identyfikację $a_{śr}$, $b_{śr}$ oraz współczynnika zajętości procesora. Należy teraz powrócić do równania ciągłości przepływu.

Solution v1

Zadanie 1 ćwiczenie 3

$J = 50$

~~faza ...~~ $a_{30} = \frac{2}{3}\,s$

$b_{30} = 80\% \cdot 1000B + 20\% \cdot 160B = 800B + 32B = 832B =$
$= 6656$ bitów

$v = 1\,Mb/s$

$p_0 = 75\%$ (750 ms przełączania w każdej sekundzie)

Równanie ciągłości przepływu:

$$1 - p_0 = J \cdot \frac{1-L}{a_{30}} \tau_{30} = J(1-L)v$$

$$L = 1 - \frac{(1-p_0) \cdot a_{30}}{J \cdot \tau_{30}}$$

$$\tau_{30} = \frac{b_{30}}{v} = \frac{6656b}{1000b/s} = 0{,}006656\,s$$

$$L = 1 - \frac{0.25 \cdot \frac{2}{3}s}{50 \cdot 0.006656\,s} = 0.4992$$

Odp. Średnia frakcja utraconych zgłoszeń wynosi 0.4992.

Solution v2

3/1

$J = 50$ $\qquad$ $L = ?$

$a_m = \frac{2}{3}$ s $\qquad$ Q - brak danych

$b_1 = 1000$ b 80%

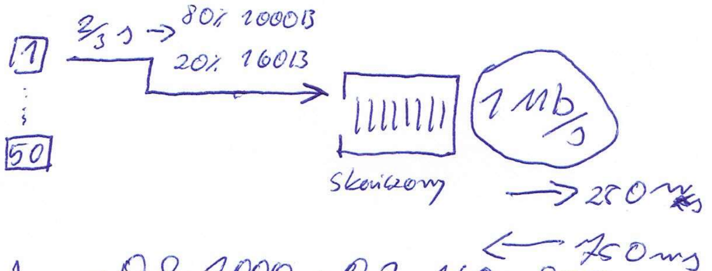
$b_2 = 160$ b 20%

V 1 Mb/s $= \frac{1\,000\,000\ b}{s}$

Pół dupleks

odbiór – niedostępny dla nas w 750 ms

250 ms - nadaje i dostępny dla nas



równanie ciągłości

$$1 - P_0 = \frac{1-L}{a_m} \quad \gamma_m = (1-L)r$$

$$b_m = 0{,}8 \cdot 1000 + 0{,}2 \cdot 160 = 832$$

$$a_m = \frac{2}{3} / 50 = \frac{2}{150}\ s \qquad V = \frac{1\,000\,000\ B}{4} = 250$$

$$r = \frac{b_m / V}{a_m} = \frac{\frac{832\ B}{1\,000\,000\ b}}{\frac{2}{150}} = \frac{6636\,b \cdot \frac{1}{1\,000\,000}}{\frac{1}{75}} = 0{,}4992\ 1/s$$

$P_0 \rightarrow$ Zakładamy, że w 250 ms procesor prawie nigdy nie będzie bezczynny bo Q duże i r duże $\rightarrow$ bezczynność ~~znaczna~~ znikoma

$P_0 = 750$ ms

$$1 - 0{,}75 = 0{,}4992 - L \cdot 0{,}4992$$
$$0{,}25 = 0{,}4992 - 0{,}4992L$$
$$0{,}2492 = 0{,}4992L$$
$$L \approx 50\%$$

Bufor bez znaczenia

Najgorsze jest grupowanie zgłoszeń

obsługa grupowa

Zamiast 2x 1 Mb/s lepiej niż 1x 2 Mb/s

Solution v3

Zad. 4

$\lambda = 50$

$a_{śr} = \frac{1}{50}$

$b_{śr} = (80\% \cdot 1000$ bajtów $+ 20\% \cdot 1600 B) \cdot 8 = 6656 b$ w bitach

$r = \frac{T}{a_{śr}}$

$r = \frac{b_{śr}}{a_{śr}}$

$1\ Mb/s = V$

$p_0 = \frac{3}{4}$ (... 750 ms) $\Rightarrow$

$1 - p_0 = \frac{1-L}{a_{śr}} \cdot T_{śr}$

$\frac{(1-p_0) \cdot a_{śr}}{T_{śr}} = 1 - L$

$L = 1 - \frac{(1-p_0)\, a_{śr} \cdot V}{b_{śr}}$

$\cancel{L = 1 - \frac{(1 - \frac{3}{4}) \cdot \frac{3}{5}}{(\frac{80}{100} \cdot 1000 + \frac{20}{100} \cdot 1600) \cdot 8 \cdot 50}}$

$L = 1 - \frac{(1-p_0) \cdot a_{śr} \cdot V}{b_{śr}}$

$L = 1 - \frac{(1 - \frac{3}{4}) \cdot \frac{4}{5} \cdot 10^6}{6656}$

Solution v4

zad 4

wymyśl $2/3$ s

odp:

80% 1000B

20% 160 B

Procek 1Mbit/s $= V$

W każdej sek 750 ms na odbiór $= p_0 = \frac{3}{4}$ (bezczynność)

$$1 - p_0 = \frac{1-L}{a_{śr}} \cdot T_{śr} \qquad T_{śr} = \frac{b_{śr}}{V}$$

$$b_{śr} = (0{,}8 \cdot 1000 + 0{,}2 \cdot 160) \cdot 8 \frac{\text{bit}}{\text{bajt}} = 6656 \text{ bitów}$$

↑ 1 końcówka

$$a_{śr} = \frac{2}{3} \cdot \frac{1}{50 \text{ końcówek}} = \frac{2}{150} \text{ s}$$

$$1 - p_0 = \frac{1-L}{a_{śr}} \cdot \frac{b_{śr}}{V} \Rightarrow L = 1 - \frac{(1-p_0) a_{śr} \cdot V}{b_{śr}} = 1 - \frac{(1-\frac{3}{4}) \cdot \frac{2 \cdot 10^6}{150}}{6656}$$

$$= 1 - \frac{10^4}{3 \cdot 6656} \approx 1 - \frac{10\,000}{20\,000} \approx 1/2$$ zgłoszeń utraconych

1 końcówka:

$$r_1 = \frac{b_{śr}}{a_{śr} \cdot V \cdot (1-p_0)} = \frac{6656}{\frac{2}{3} \cdot 0{,}25 \cdot 10^6} \approx 4‰$$ obciążenia

Ex 3.2, Old ex 5

Problem

Problem 2

In a single-processor queuing system with buffer capacity $Q = 2$ in statistical equilibrium and under offered load $r = 0.75$, we have $p_0 \geq p_1 \geq p_2$. Find the range of possible p_1 values.

Proceed as in the previous problem.

Zadanie 5

O 1-procesorowym systemie masowej obsługi z pamięcią buforową o pojemności $Q = 2$ wiadomo, że w stanie ustalonym mamy $p_0 \geq p_1 \geq p_2$ oraz że $r = 0.75$. Znajdź przedział możliwych wartości p_1.

Należy postąpić tak, jak w poprzednim zadaniu.

Zadanie 2

O 1-procesorowym systemie masowej obsługi z pamięcią buforową o pojemności $Q = 2$ wiadomo, że w stanie ustalonym mamy $p_0 \geq p_1 \geq p_2$ oraz że $r = 0.75$. Znajdź przedział możliwych wartości p_1.

Należy postąpić tak, jak w poprzednim zadaniu.

Solution v1

Zadanie 2 ćwiczenia 3

$Q = 2$

$p_0 \geq p_1 \geq p_2$

$v = 0.75$

$$(1-p_0)+(1-p_1)+(1-p_2) = Q$$

$$\frac{1-p_0}{1-p_2} = 0.75 = \frac{3}{4}$$

$1-p_0$	$1-p_1$	$1-p_2$
3	x	4

$$\frac{1-p_1}{2} = \frac{x}{3+x+4} \Rightarrow p_1 = 1 - \frac{2x}{x+7}$$

$$p_0 \geq p_1 \geq p_2 \Rightarrow (1-p_0) \leq (1-p_1) \leq (1-p_2)$$

$$x \in (3, 4)$$

dla $x = 3$:

$$p_1 = 1 - \frac{2 \cdot 3}{3+7} = 1 - \frac{6}{10} = \frac{4}{10} = \frac{2}{5}$$

dla $x = 4$:

$$p_1 = 1 - \frac{2 \cdot 4}{4+7} = 1 - \frac{8}{11} = \frac{3}{11}$$

$$p_1 \in \left(\frac{3}{11}, \frac{2}{5}\right)$$

$$p_0 + p_1 + p_2 = 1$$

$$1 - p_0 = \frac{1-L}{Q_{sv}} \cdot \tau_{sv} = (1-L)v$$

$L = p_2$

$$1 - p_0 = (1-p_2) \cdot 0.75$$

5

Solution v2

Zad. 5 $Q=2$ pamięć buforowa $p_0 \geq p_1 \geq p_2$ $r=0{,}75$ obciążenie procesora

$p_1 = ?$

$p_0 + p_1 + p_2 = 1$

$1 - p_0 = (1-L)\cdot r$ $r = 0{,}75$

$L = p_2$

$p_2 = L$ traci zgłoszenie wejście, bo $Q=2$

p_2 = 2 zgł. w systemie = prawdop. że zgłoszenie które przyjdzie, zastanie 2 zgł. w systemie

p_1 = 1 zgł. w systemie

$$\begin{cases} 1 - p_0 = (1-p_2)\cdot r \\ p_0 + p_1 + p_2 = 1 \end{cases}$$

$$\begin{cases} 1 - p_0 = (1-p_2)\cdot \frac{3}{4} \\ 1 - p_0 = p_1 + p_2 \end{cases}$$

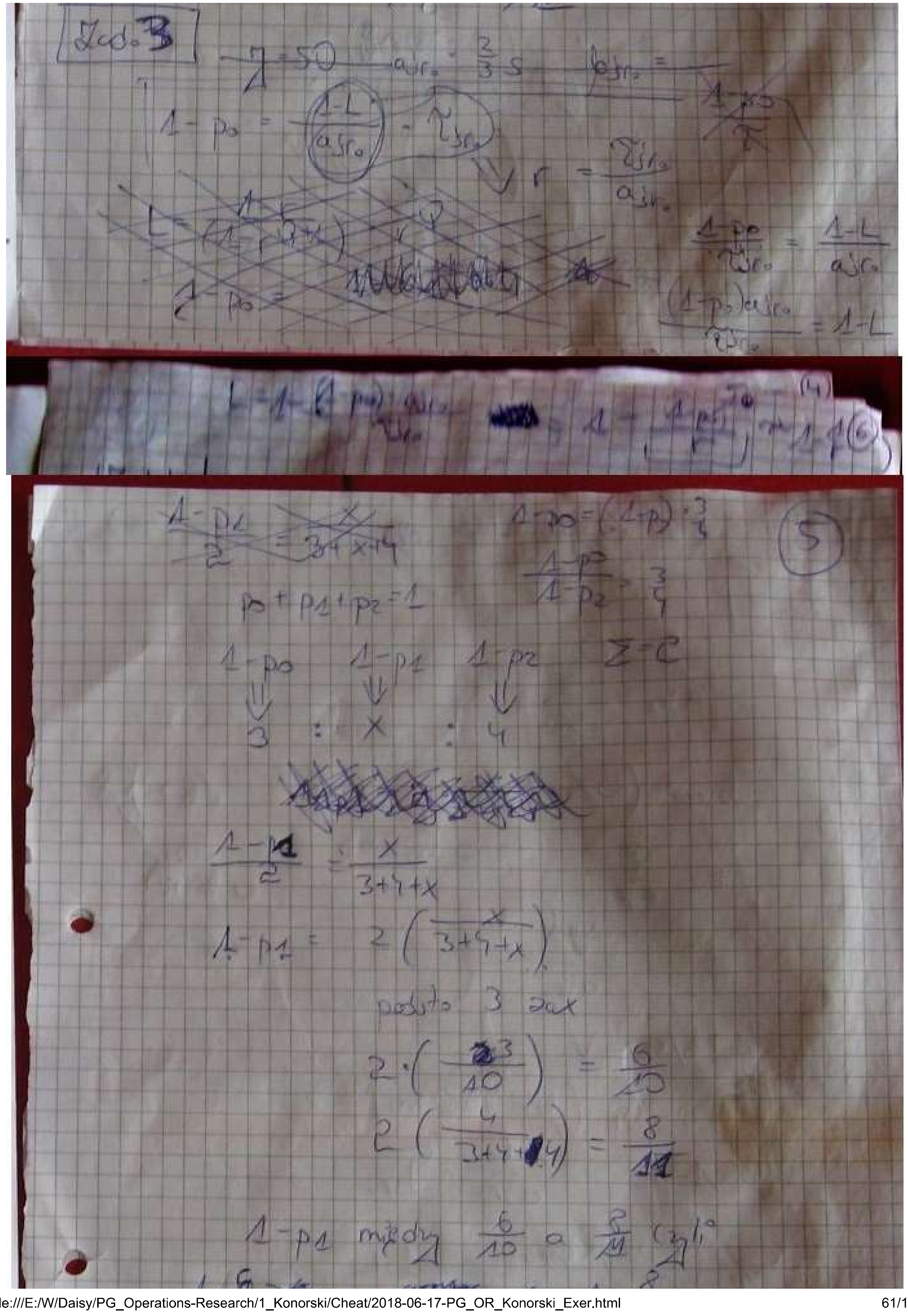
$$(1-p_2)\frac{3}{4} = p_1 + p_2$$

$$p_1 + p_2 = \frac{3}{4} - \frac{3}{4}p_2$$

$$p_1 = \frac{3}{4} - 1\frac{3}{4}p_2$$

$$1 - p_0 + 1 - p_2 + 1 - p_1 = 2$$

Solution v3


Zad. 3
λ = 50
a_śr = 2/3 s
1 - p0 = (1-L)/a_śr · τ_śr
r = τ_śr / a_śr
(1-p0)/τ_śr = (1-L)/a_śr
(1-p0) a_śr / τ_śr = 1-L
L = 1 - (1-p0) a_śr / τ_śr
5
1-p0 = (1-p2) · 3/4
p0 + p1 + p2 = 1
1-p0 1-p1 1-p2 Σ = 2
3 : x : 4
(1-p1)/2 = x/(3+4+x)
1-p1 = 2(x/(3+4+x))
podstaw 3 za x
2·(3/10) = 6/10
2(4/(3+4+4)) = 8/11
1-p1 między 6/10 a 8/11 czyli

$1 - \frac{[illegible]}{10} \geq p_1 \geq 1 - \frac{[illegible]}{11}$

$\frac{4}{10} \geq p_1 \geq \frac{3}{11}$

Solution v4

zad 5

$Q = 2 \qquad p_1 = ?$

$r = 0{,}75$

$(1-p_0) + (1-p_1) + (1-p_2) = \underline{\underline{2}}$ (całość)

$$\frac{1-p_0}{1-p_2} = 0{,}75 = \frac{3}{4}$$

$1-p_0$	$1-p_1$	$1-p_2$
3	X	4

$$\frac{1-p_1}{2} = \frac{x}{3+x+4}$$

$x \in (3, 4)$

$$1-p_1 = \frac{2x}{x+7} \qquad p_1 = 1 - \frac{2x}{x+7}$$

dla $x=3$

$$p_1 = 1 - \frac{2 \cdot 3}{7+3} = \frac{4}{10}$$

dla $x=4$

$$p_1 = 1 - \frac{2 \cdot 4}{7+4} = \frac{3}{11}$$

$$p_1 \in \left(\frac{3}{11}, \frac{4}{10}\right)$$

$p_0 + p_1 + p_2 = 1$

$$1-p_0 = \frac{1-L}{a_{śr}} \cdot T_{śr} \rightarrow 1-p_0 = (1-\overset{p_2}{L}) \cdot \overset{0{,}7}{r}$$

$$1-p_0 = (1-p_2) \cdot \frac{3}{4}$$

Ex 3.3, Old ex

Problem

Problem 3 (simulation experiment)

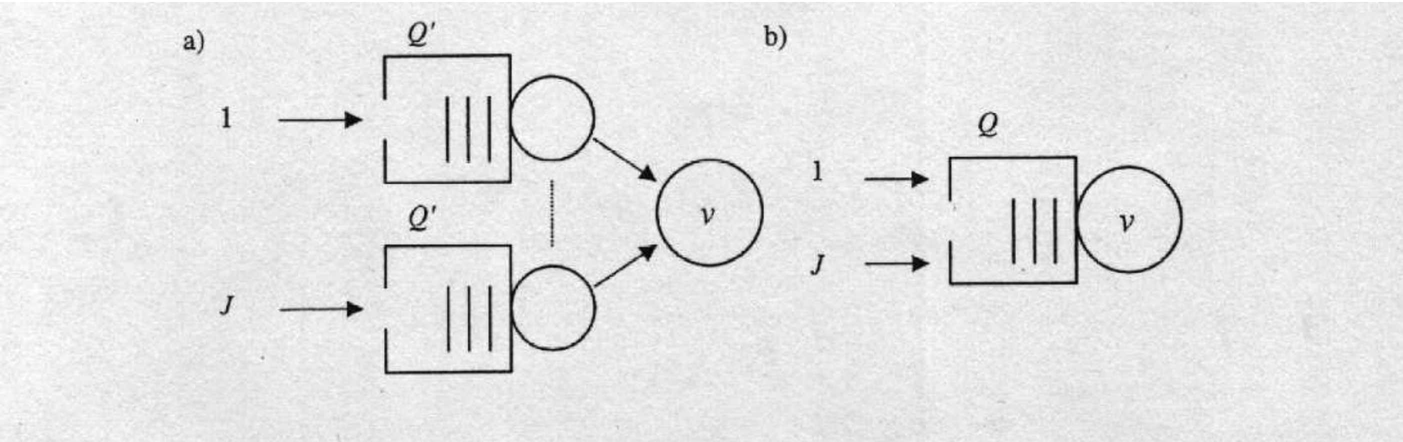
Each user of a single-processor queuing system generates an arrival stream of documents with mean interval a_m s. Documents have a mean length of b_m bytes. The processor handles documents at speed v bytes/s. Tolerable are:

- document loss fraction due to buffer overflow not exceeding L_{max}
- mean system delay of a document not exceeding a given multiple c of b_m/v.

Subject to the above, compare the maximum number J_{max} of users and required buffer capacity in two configurations:

(a) *dedicated access* with a virtual processor and a separate buffer space assigned for each user, and
(b) *common-buffer access* to the processor.

Perform simulations for $a_m = 6$ s, $b_m = 600$ bytes, $v = 24000$ bytes/s, $L_{max} = 0.1\%$, $c = 5$.

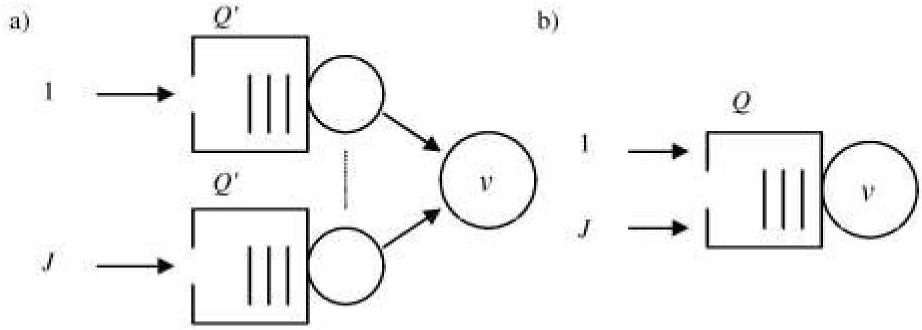


Zadanie 3 (eksperyment symulacyjny)

Każdy spośród użytkowników systemu jednoprocesorowego generuje strumień dokumentów ze średnim interwałem $a_{śr}$. Długości dokumentów mają średnią $b_{śr}$. Wydajność przetwarzania dokumentów przez procesor wynosi v. Dopuszcza się:

- frakcję dokumentów utraconych wskutek przepełnienia nie większą niż L_{max},
- średnie opóźnienie systemowe dokumentu nie większe niż c-krotność $b_{śr}/v$.

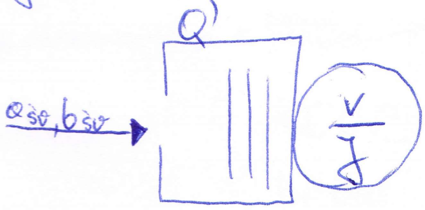
Porównaj maksymalną liczbę J_{max} użytkowników oraz wymaganą pojemność pamięci buforowych dla przypadków: a) *dostęp dedykowany* z procesorami wirtualnymi o jednakowej wydajności i z odrębnymi kolejkami, b) *dostęp ze wspólną kolejką* do procesora fizycznego. Wykonaj symulacje dla $a_{śr} = 6$ s, $b_{śr} = 0.6$ KB, $v = 24$ KB/s, $L_{max} = 0.1\%$, $c = 5$.


a)
Q'
1
Q'
J
v
b)
Q
1
J
v

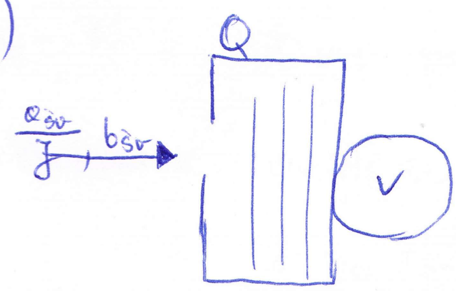
Solution v1

Zadanie 5 ćwiczenie 3

a) System można interpretować w ten sposób:



b)



W obu przypadkach wykorzystując przykładowe dane należy obliczyć J_{max} oraz Q (lub Q') dla takiej liczby użytkowników.

$$\left.\begin{array}{l} L \leq L_{max} \\ d_{30} \leq c \cdot \dfrac{b_{30}}{v} \\ \rho = \dfrac{b_{30}}{a_{30} \cdot v} \\ L = \begin{cases} \dfrac{1-\rho}{1-\rho^{Q+1}} \cdot \rho^{Q} & \text{dla } \rho \neq 1 \\ \dfrac{1}{Q+1} & \text{dla } \rho = 1 \end{cases} \\ d_{30} = \dfrac{N_{30}}{\frac{1-L}{a_{30}}} \end{array}\right\} \text{system } M/M/1/$$

Ex 3.Extra, Old ex 9

Problem

Zadanie 9

Do 1-procesorowego systemu masowej obsługi przybywa strumień niecierpliwych zgłoszeń ze średnim interwałem $a_{śr}$ i średnim wymaganiem $b_{śr}$. Wydajność procesora wynosi v. W chwili przybycia zgłoszenie określa swój indywidualny "próg cierpliwości" (średnia wartość wynosi $c_{śr}$). Jeżeli po jego upływie ciągle jeszcze oczekuje na rozpoczęcie obsługi, ucieka z kolejki. W przeciwnym razie opuszcza system po zakończeniu obsługi. a) Udało nam się pomierzyć współczynnik bezczynności p_0 procesora – jak stąd wyznaczyć frakcję L zgłoszeń, które uciekły z kolejki? b) Udało się też pomierzyć średnią liczbę $N_{śr}$ zgłoszeń w systemie – jak stąd wyznaczyć średnie opóźnienie buforowania zgłoszeń, których obsługa została zrealizowana?

Przedyskutuj zastosowanie prawa Little'a. W punkcie b) ostrożnie!

$N_{śr}/[(1-L)/a_{śr}] - b_{śr}/v$?

$\ldots = (1-L)X + L\,c_{śr}$?

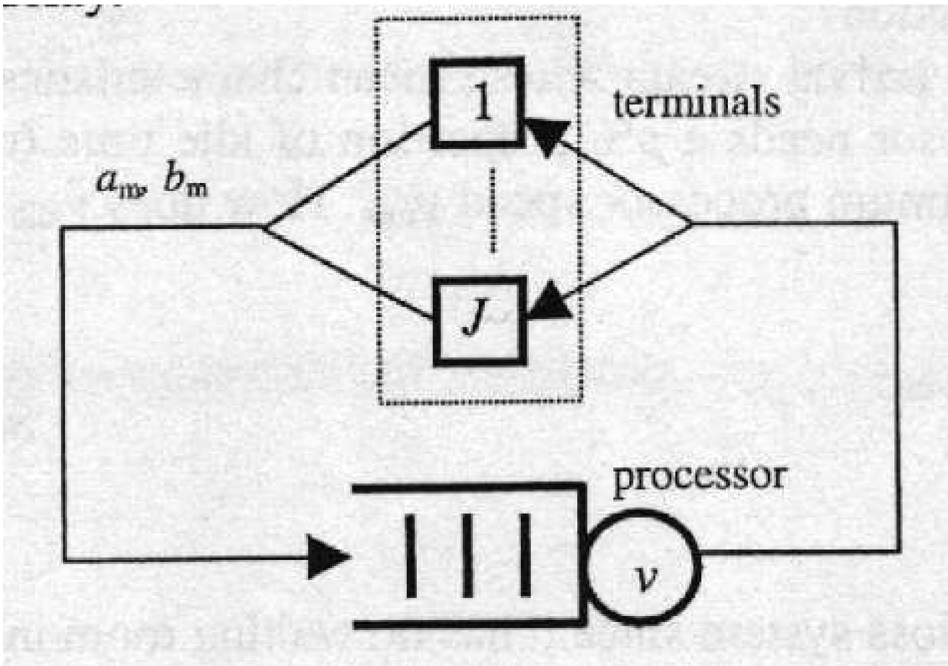
Solution

Ex 4.1, Old ex 6

Problem

Problem 1

A single-processor queuing system interacts with $J = 10$ intelligent terminals in a query-response manner, as depicted below. Having received a response, a terminal generates a new query after a think time of average duration $h_m = 4$ s. The average number of elementary operations needed to generate a response is $b_m = 15000$, and processor is $v = 5000$ elementary operations/s. Find the relationship between the proportion of processor idle time and mean *waiting* delay.

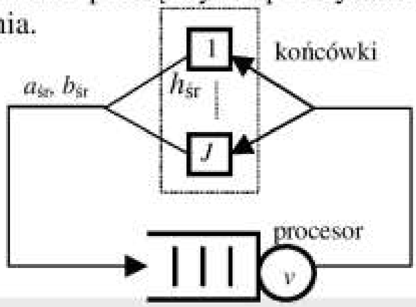


Hint

The flow conservation equation again comes in handy.

Zadanie 1

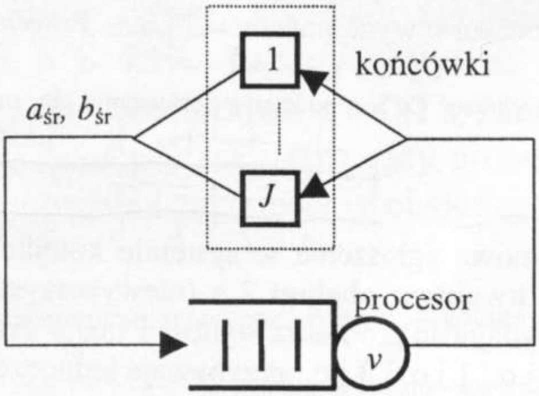
1-procesorowy system masowej obsługi współpracuje z $J = 10$ inteligentnymi końcówkami w trybie konwersacyjnym zapytanie-odpowiedź. Po otrzymaniu odpowiedzi końcówka generuje nowe zapytanie po czasie namysłu wynoszącym średnio $h_{śr} = 4$ s. Średnia liczba operacji niezbędnych do wygenerowania odpowiedzi wynosi $b_{śr} = 15000$, zaś procesor w systemie posiada wydajność $v = 5000$ operacji/s. Wyznacz zależność pomiędzy współczynnikiem bezczynności procesora a średnim opóźnieniem buforowania.



Przyda się prawo Little'a.

Zadanie 6

1-procesorowy system masowej obsługi współpracuje z $J = 10$ inteligentnymi końcówkami w trybie konwersacyjnym zapytanie-odpowiedź. Po otrzymaniu odpowiedzi końcówka generuje nowe zapytanie po czasie namysłu wynoszącym średnio $a_{śr} = 4$ s. Średnia liczba operacji niezbędnych do wygenerowania odpowiedzi wynosi $b_{śr} = 15000$, zaś procesor w systemie posiada wydajność $v = 5000$ operacji/s. Wyznacz zależność pomiędzy współczynnikiem bezczynności procesora a średnim opóźnieniem buforowania.



Znów przyda się równanie ciągłości przepływu.

Solution v1

Zadanie 1 ćwiczenia 4

Prawo Little'a:

$$\overline{\text{populacja}} = \overline{\text{cyrkulacja}} \times \overline{\text{czas życia}}$$

$\overline{\text{populacja}}$ = J zgłoszeń = 10

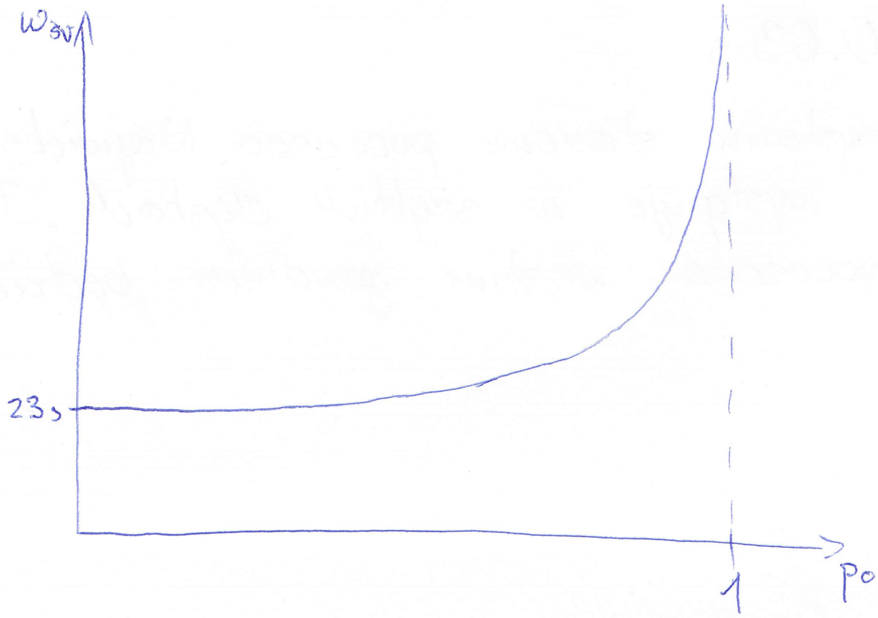
$\overline{\text{czas życia}} = h_{ŚV} + \frac{b_{ŚV}}{V} + w_{ŚV} = 4s + \frac{15000}{500}s + w_{ŚV} = 7s + w_{ŚV}$

$\overline{\text{cyrkulacja}} = \frac{1}{b_{ŚV}/V} \cdot (1-p_0) = \frac{1}{a_{ŚV}}$

$$10 = \frac{1-p_0}{\frac{b_{ŚV}}{V}} \times (7s + w_{ŚV}) \qquad \frac{b_{ŚV}}{V} \to 3s$$

$$30s = (1-p_0) \times (7s + w_{ŚV})$$

$$w_{ŚV} = \frac{30s}{1-p_0} - 7s \quad \leftarrow \text{szukana zależność}$$



6

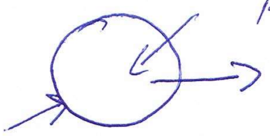
Solution v2

4/1

$b_m = 15\,000$

$V = 5\,000$

$J = 10$



prawo Little

$\overline{\text{populacja}} = \overline{\text{cyrkulacja}} \times \overline{\text{czas życia}}$

mój system do którego stosuję prawo Little

Stosuję to do całego systemu

Cyrkulacja wiele na poszczególnych końcówkach

populacja → J zapytań, bo każda końcówka generuje 1 zapytanie

Czas życia $= \frac{b_m}{V} + w_m + h_m$ ← czas namysłu

$J = \frac{1}{h_m + w_m + \frac{b_m}{V}}$ — co tyle czasu generuje końcówka zapytanie

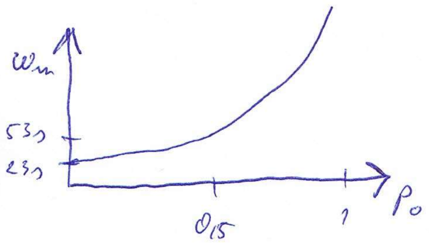
Prędkość cyrkulacji w układzie zamkniętym to zawsze $\frac{1}{a_m}$

Na cpu jeszcze raz prawo Little → na cpu, czyli równanie przepływu

$1-P_0 = (1-L)\nu \longrightarrow 1-P_0 = \nu \longrightarrow 1-P_0 = \frac{b_m/V}{a_m} \longrightarrow \frac{1}{a_m} = \frac{1-P_0}{b_m V}$

bufor nieskończony

$J = \frac{1-P_0}{b_m/V}\left(h_m + w_m + \frac{b_m}{V}\right), \quad w_m = \frac{J \cdot \frac{b_m}{V}}{1-P_0} - h_m - \frac{b_m}{V}$



Solution v3

Ćw. 4 1) $N = 10$ $h = 4s$ $b_{śr} = 15000$ $V = 5000$

p_0 = wsp. bezczynności proca

$w_{śr}$ = średnie opóźnienie buforowania (ilość czasu stracona w kolejce)

$$w_{śr} = \frac{N \cdot b_{śr}/V}{1-p_0} - h_{śr} - \frac{b_{śr}}{V}$$ (3)

$p_0 = ?$ $a_{śr} = ?$ $a_{śr} =$

$$\boxed{N_{śr} = \frac{1-L}{a_{śr}} \cdot d_{śr}} \Leftarrow \text{prawo Little'a}$$

$N_{śr}$ – populacja (ilość zgłoszeń przebywających w systemie) średnia liczba transakcji

$d_{śr}$ – czas przebywania zgłoszenia w systemie

po przekształceniu:

$$\frac{1-L}{a_{śr}} = \frac{1-p_0}{\tau}$$

$\tau_{śr}$ → czas przetwarzania + czas buforowania ($w_{śr}$) + czas namysłu ($a_{śr}$)

Równanie ciągłości przepływu:

$$1-p_0 = \frac{1-L}{a_{śr}} \cdot \tau_{śr} \quad \text{lub} \quad 1-p_0 = (1-L) \cdot r$$

$N_{śr} = 10$ (bo jest 10 komputerów)

$$w_{śr} = \frac{10 \cdot 15000 \cdot \frac{1}{5000}}{1-p_0} - 4 - \frac{15000}{5000} =$$

$$\boxed{\frac{30}{1-p_0} - 7} \Rightarrow \frac{30}{1-p_0} - 7.$$

Solution v4

zad 6

$J = 10$

$V = 1 \times 5000$ op/s

$b_{śr} = 15\,000$ op

$a_{śr} = 4$ s

śr popul = śr cyrk x śr czas życia

$$T_{śr} = \frac{b_{śr}}{V} = \frac{15\,000 \text{ op}}{5000 \text{ op/s}} = 3\text{s}$$

populacja = 10

"częstotliwość"

?

$$10 = \frac{1}{T_{śr}} \cdot (1-p_0) \cdot \left(a_{śr} + w_{śr} + \frac{b_{śr}}{V}\right)$$

?

$$10 = \frac{1}{3} \cdot (1-p_0) \cdot (4 + w + 3)$$

$$\frac{10 \cdot 3}{1-p_0} = 7 + w \Rightarrow w = \frac{30}{1-p_0} - 7$$

Solution v5

4.1 | $s = 1$ $\quad J = 10 = N_{śr}$, bo końcówka generuje nowe zapytanie dopiero po otrzymaniu odpowiedzi zatem w sys jest zawsze 1 zgł. na końcówkę

$h_{śr} = 4\,s \quad b_{śr} = 15\,000;$

$v = 5000\,j/s \quad \frac{p_0}{w_{śr}} = ?$

$\tau_{śr} = 3\,s$

$J = N_{śr} = \frac{1-L}{a_{śr}}\left(w_{śr} + \tau_{śr} + h_{śr}\right)$ $\quad L = 0$, bo nic nie traci

$J = \frac{w_{śr} + \tau_{śr} + h_{śr}}{a_{śr}}$ $\quad \& \quad a_{śr} = (1-p_0)\cdot\frac{1}{\tau_{śr}}$

przepływność przetwarzania × % czasu który pracuje = cyklicznego

$J = \frac{(w_{śr} + \tau_{śr} + h_{śr})\,\tau_{śr}}{(1-p_0)}$

$(1-p_0)J = w_{śr}\tau_{śr} + \tau_{śr}^2 + h_{śr}\tau_{śr} \Rightarrow$

$J - p_0 J = w_{śr}\tau_{śr} + \tau_{śr}^2 + h_{śr}\tau_{śr} \Rightarrow$

$10 - 10p_0 = 3w_{śr} + 9 + 12 \Rightarrow 10 - 21 = 3w_{śr} + 10p_0$

$-11 = 3w_{śr} + 10p_0 \quad -3w_{śr} = 10p_0 + 11 \quad \boxed{w_{śr} = \frac{10p_0 + 11}{-3}}$

Ex 4.2, Old ex 8

Problem

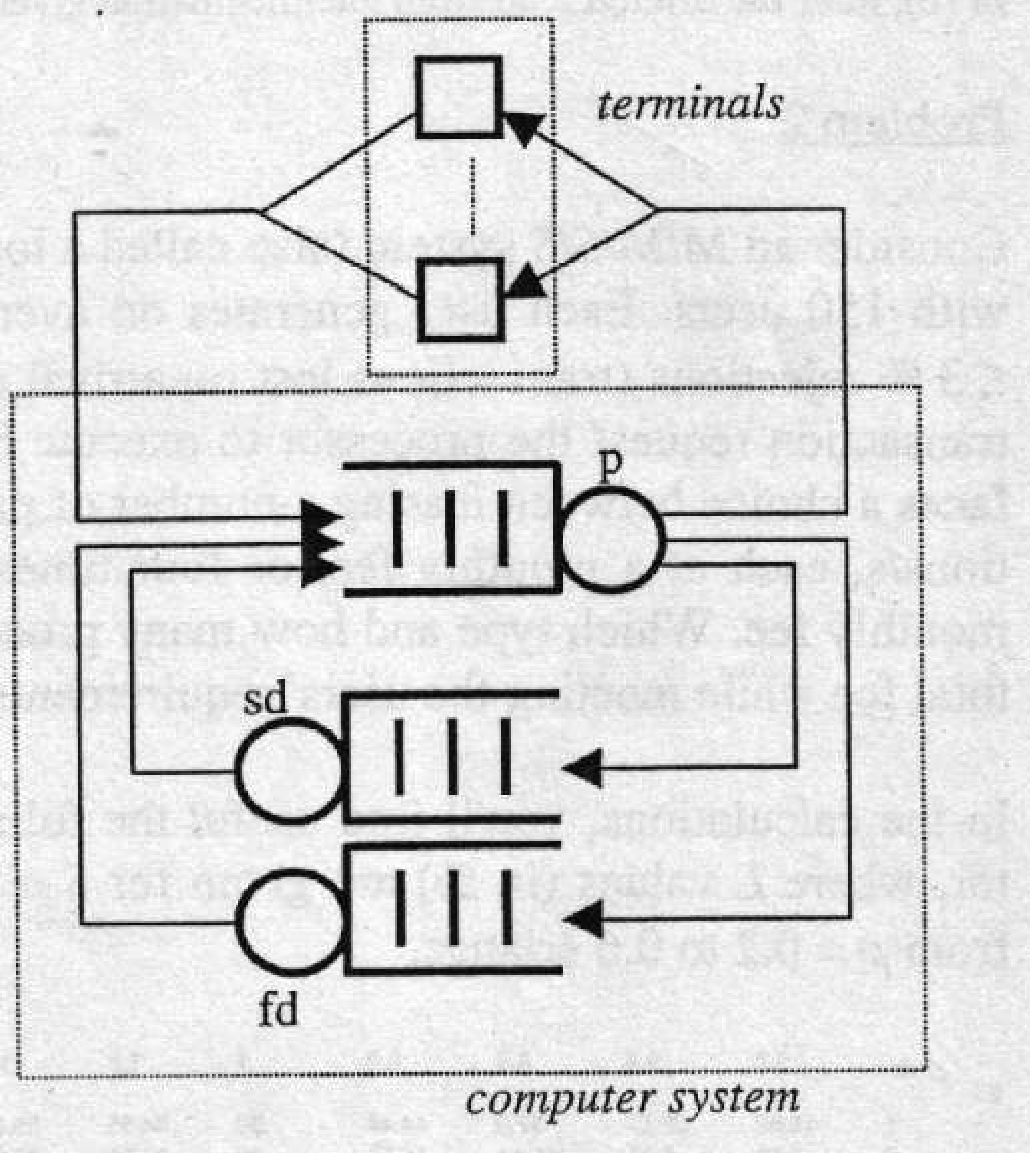
Problem 2

computer system

Each of $J = 30$ computer terminals generates requests that require sequential service at a processor, slow-disk controller and fast-disk controller, as depicted below. On average, a request has to visit these devices $l_p = 21$, $l_{sd} = 12$, $l_{fd} = 8$ times, respectively, whereas average service times there equal $\tau_p = 0.05$ s, $\tau_{sd} = 0.07$ s and $\tau_{fd} = 0.02$ s. Upon notification of service completion for its request, a terminal enters a think time of average $h_m = 15$ s, and subsequently generates another request.

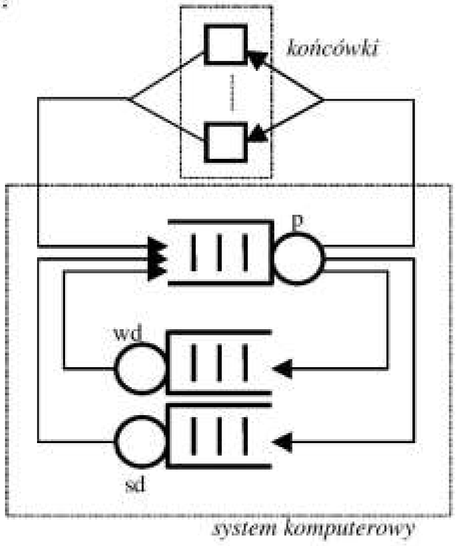
(a) Which device is the bottleneck, and which one is the most overdimensioned? How will his change if the processor is tuned up so that $\tau_p = 0.03$ s?
(b) What processor speedup do we need in order for the mean system delay (request time within the system) to become $d^*_m = 12$ s, and what speedup would ensure $d^*_m = 9$ s?

Assume a certain request arrival interval at the terminal-system interface. Use it to express the offered load at each device. For (b), will Little's theorem be of any use?


terminals
p
sd
fd
computer system

Zadanie 2

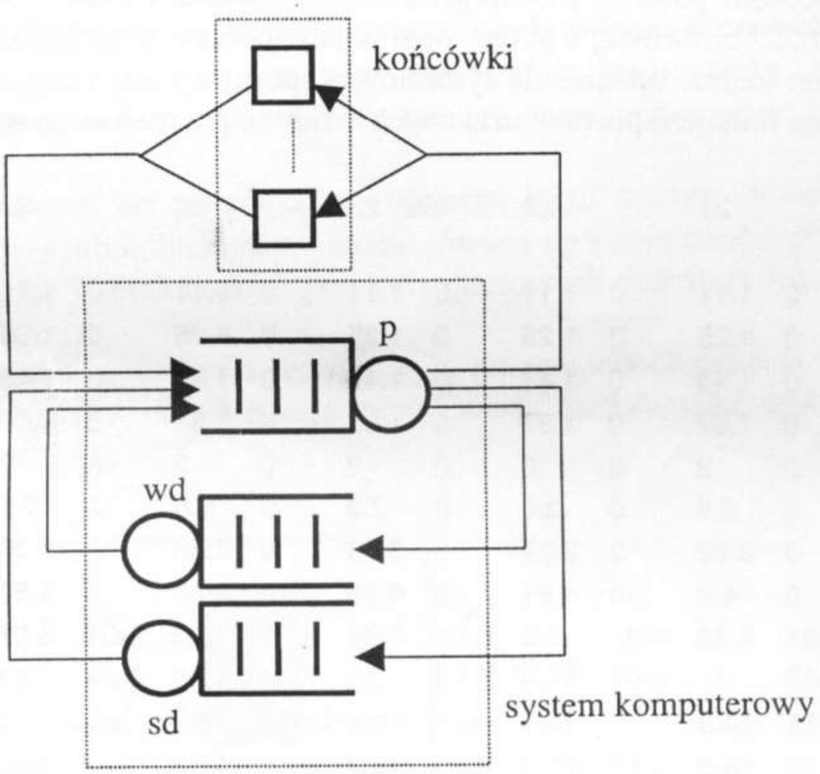
Każda z J = 30 końcówek systemu komputerowego generuje zgłoszenia wymagające sekwencyjnego przetwarzania w procesorze, stacji wolnych dysków i stacji szybkich dysków. Średnie liczby wizyt zgłoszenia w urządzeniach wynoszą $l_p = 21$, $l_{wd} = 12$, $l_{sd} = 8$, zaś średnie czasy przetwarzania w trakcie wizyty wynoszą $\tau_p = 0.05$ s, $\tau_{wd} = 0.07$ s, $\tau_{sd} = 0.02$ s. Po zakończeniu przetwarzania zgłoszenia końcówka przechodzi do fazy namysłu trwającej średnio $h_{śr} = 15$ s, po czym generuje kolejne zgłoszenie.
a) Gdzie jest wąskie gardło systemu, a gdzie występuje największe przewymiarowanie? Jak się to zmieni, gdy przyśpieszymy procesor tak, że $\tau_p = 0.03$ s?
b) Jakie przyśpieszenie procesora niezbędne jest dla uzyskania średniego opóźnienia systemowego $d^*_{śr} = 12$ s, a jakie dla uzyskania $d^*_{śr} = 9$ s?



Załóż jakąś prędkość przepływu [zgł/s] na styku systemu komputerowego i zbioru końcówek. Wyraź przy jego pomocy współczynnik obciążenia dowolnego urządzenia w systemie. W punkcie b) wykorzystaj prawo Little'a.

Zadanie 8

Każda z J = 30 końcówek systemu komputerowego generuje zgłoszenia wymagające sekwencyjnego przetwarzania w procesorze, stacji wolnych dysków i stacji szybkich dysków – rysunek. Średnie liczby wizyt zgłoszenia w urządzeniach wynoszą $l_p = 21$, $l_{wd} = 12$, $l_{sd} = 8$, zaś średnie czasy przetwarzania w trakcie wizyty wynoszą $\tau_p = 0.05$ s, $\tau_{wd} = 0.07$ s, $\tau_{sd} = 0.02$ s. Po zakończeniu przetwarzania zgłoszenia końcówka przechodzi do fazy namysłu trwającej średnio $a_{śr} = 15$ s, po czym generuje kolejne zgłoszenie. a) Gdzie jest wąskie gardło systemu, a gdzie występuje największe przewymiarowanie? Jak się to zmieni, gdy przyśpieszymy procesor tak, że $\tau_p = 0.03$ s? b) Jakie przyśpieszenie procesora niezbędne jest dla uzyskania średniego opóźnienia systemowego (czasu przebywania zgłoszenia w systemie komputerowym) $d^*_{śr} = 12$ s, a jakie dla uzyskania $d^*_{śr} = 9$ s?



Załóż jakąś prędkość przepływu [zgł/s] na styku systemu komputerowego i zbioru końcówek. Wyraź przy jej pomocy współczynnik obciążenia dowolnego urządzenia w systemie. Czy uda się wykorzystać prawo Little'a?

Solution v1: A), B)

Zadanie 2 ćwiczenia 4

$l_p = 21$ $\tau_p = 0.05\,s$ → procesor

$l_{wd} = 12$ $\tau_{wd} = 0.07\,s$ → wolne dyski

$l_{sd} = 8$ $\tau_{sd} = 0.02\,s$ → szybkie dyski

$J = 30$

$a_{św} = 15\,s$

$$v_x = \frac{\tau_x}{a_{św}/l_x} = \frac{1}{a_{św}} \cdot \tau_x \cdot l_x$$

takie same ← porównać

$\tau_p \cdot l_p = 1.05\,s$

$\tau_{wd} \cdot l_{wd} = 0.84\,s$

$\tau_{sd} \cdot l_{sd} = 0.16\,s$

po przyspieszeniu procesora ($\tau_p = 0.03\,s$)

$\tau_p \cdot l_p = 0.63\,s$

a) wąskie gardło systemu stanowi procesor. Największe przewymiarowanie występuje w szybkich dyskach. Po przyspieszeniu procesora wąskim gardłem będzie wolny dysk.

b)

$$\overline{\text{populacja}} = \overline{\text{cyrkulacja}} \times \overline{\text{czas życia}}$$

$$J = (h_{św} + d_{św}) \times \frac{1}{a_{św}} \Rightarrow a_{św} = \frac{h_{św} + d_{św}}{J}$$

$d_{św} = 12\,s$

$a_{św} = \frac{15 + 12}{30} = 0.9$

$\frac{1}{a_{św}} = 1.11$ ← cyrkulacja, którą trzeba wymusić

$v_p = 1.11 \cdot 1.05 = 1.166$ ← czy taką cyrkulację wytrzyma procesor!

Odp. Nie. Jest to ponad 116% możliwości procesora.

Solution v2: A), B)

4/2

a)

$$CPU = \frac{L_p \cdot T_p}{a_m} = \frac{21 \cdot 0{,}05}{a_m} = \frac{1{,}05}{a_m}$$ ← wąskie gardło

a_m - na wejście systemu komputerowego

$$Sd = \frac{L_{sd} \cdot T_{sd}}{a_m} = \frac{12 \cdot 0{,}07}{a_m} = \frac{0{,}84}{a_m}$$

a_m - względnie to samo

$$Fd = \frac{L_{fd} \cdot T_{fd}}{a_m} = \frac{0{,}16}{a_m}$$

$CPU_v \leq 1$ $\quad \frac{1{,}05}{a_m} \leq 1$

$$\frac{1}{a_{m\,min}} \approx \frac{1}{1{,}05}$$

Cyrkulacja maksymalna

$Fd \rightarrow \frac{0{,}16}{1{,}05} \approx 15{,}6\%$ tego wykorzystanie nie przekroczymy, reszta się marnuje

b) ograniczyć opóźnienie systemowe nie zajęTO więcej niż...

problem z czasem kolejkowania

znamy tylko czas obsługi

Prawo Little - najpierw całości

$$T = \frac{1}{a_m} \cdot (h_m + d_m^*)$$

stałe

nie zmniejszenie...

czas zwłoki → max 12 s

Trzeba przyspieszyć cyrkulację ...

Powinno wyjść w odpowiedzi, że nie wytrzyma

a gdy d = 9 ?

Solution v3

A)

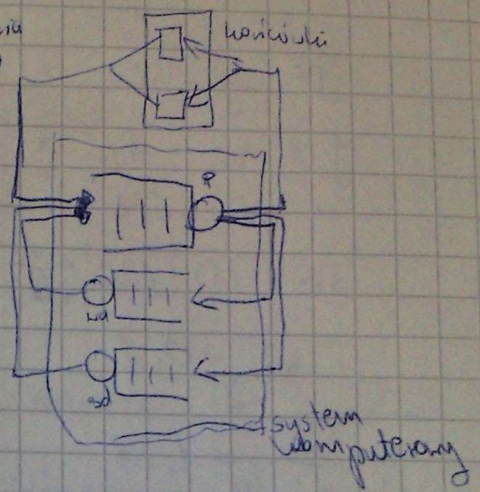
Badania operacyjne ***, W, 07.03.2012 r.

zestaw IV

Zadanie 2. $J = 30$ końcówek

wymagana liczba wizyt w zgłoszeniach

czasy przetwarzania podczas wizyty



$l_p = 21$ $\quad \tau_p = 0{,}05$

$l_{wd} = 12$ $\quad \tau_{wd} = 0{,}07$

$l_{sd} = 8$ $\quad \tau_{sd} = 0{,}02$

faza namysłu $a_{śr} = 15s$

$J = 30$

$h_{śr} = 15s$

Wąskie gardło jest tam, gdzie jest największy wsp. obciążenia

$$r = \frac{\tau_{śr}}{a_{śr}}$$

$\tau_{śr}$ ← wymagany średni czas obsługi

$a_{śr}$ ← interwał między zgłoszeniami

Nie znamy $a_{śr}$, ale jest takie samo dla wszystkich urządzeń, więc zadecyduje licznik

$$\left.\begin{aligned} r_p &= \frac{21 \cdot 0{,}05}{a_{śr}} \\ r_{wd} &= \frac{12 \cdot 0{,}07}{a_{śr}} \\ r_{sd} &= \frac{8 \cdot 0{,}02}{a_{śr}} \end{aligned}\right\} , 60$$

$\underline{\tau_{śr} = l \cdot \tau}$ (liczba wizyt × czas wizyty)

a)

$r_p = \frac{1{,}05}{a_{śr}}$ → wąskie gardło $\quad r_{pp} = \frac{21 \cdot 0{,}03}{a_{śr}} = \frac{0{,}63}{a_{śr}}$

$r_{wd} = \frac{0{,}84}{a_{śr}}$

$r_{sd} = \frac{0{,}16}{a_{śr}}$

Po przyspieszeniu procesora wąskim gardłem staje się WD.

B)

b)

$$N_{śr} = \frac{1-L}{a'_{śr}} \cdot d_r$$

$$1 - p_0 = \frac{1-L}{a_{śr}} \tau_{śr} = (1-L) r$$

J = cyrkulacja × średni czas życia

$$J = h_{śr} + d_{śr} * \frac{1}{a'_{śr}}$$

$30 \quad 15 \quad \leq 12$

$$30 = (15 + 12) * \frac{1}{a'_{śr}}$$

$$a'_{śr} = 0,9$$

$$\frac{1}{a'_{śr}} = 1,11$$

$$r_p = 1,11 \cdot \frac{\text{zał.}}{3} \cdot 1,05_s = 1,166$$

$$r_p = \frac{1}{a'_s} \cdot L_p \cdot \tau_p$$

$$\frac{0,05}{1,166}$$ → nowy średni czas wizyty

Ex 5.1, Old ex

Problem

Problem 1

Answer the following questions related to finite-buffer queuing systems.

(a) An average of 40 requests per second arrive in an M/M/1/2 queuing system, each requesting 20 ms of processor time. How many requests on average are lost per day due to buffer overflow?
(b) In an M/M/1/5 queuing system, two requests arrive on average during the service of a request of average length. What is the loss fraction?
(c) An M/M/1/Q queuing system serves an arrival stream whose mean characteristics are a_m and b_m. For sustained operation, the processor needs a $p\%$ proportion of idle time (used for maintenance), which in turn requires a minimum processor speed v_{min}. How does v_{min} change with increasing Q?

In (b), infer the offered load from the information given.

Zadanie 1

a) W systemie M/M/1/2 średnia intensywność przybywania zgłoszeń wynosi 40 na sekundę, zaś średni czas obsługi zgłoszenia wynosi 20 ms. Ile zgłoszeń średnio zostaje utraconych w ciągu doby?

b) W systemie M/M/1/5 w trakcie obsługi zgłoszenia o średnim wymaganiu obsługi przybywają średnio dwa nowe zgłoszenia. Jak duża jest frakcja zgłoszeń utraconych wskutek przepełnienia pamięci buforowej?

c) W systemie M/M/1/Q strumień zgłoszeń ma dane parametry $a_{śr}$ i $b_{śr}$. Dla prawidłowej pracy procesor potrzebuje p-% udziału czasu bezczynności, do czego z kolei wymagana jest pewna minimalna wydajność obsługi v_{min}. Jak v_{min} zmienia się ze wzrostem Q?

Informację zawartą w punkcie b) przetłumacz na współczynnik obciążenia.

Solution v1

A), B)

Zadanie 1 ćwiczenia 5

a) M|M|1|2 ← pojemność bufora (Q)

liczba procesorów (v)

średnia intensywność ($a_{śr}$)

średnie wymaganie ($b_{śr}$)

$a_{śr}$ = ~~40 zgłoszeń/sekunda~~ $\frac{1}{40} s = 0.025 s$

$b_{śr} = 20 ms = 0.02 s$

$$\rho = \frac{b_{śr}}{a_{śr} \cdot v} = \frac{0.02 s}{0.025 s \cdot 1} = 0.8$$

dla $\rho \neq 1$ $\quad L = p_Q = \frac{1-\rho}{1-\rho^{Q+1}} \cdot \rho^Q$

$$L = \frac{1-0.8}{1-(0.8)^3} \cdot (0.8)^2 = \frac{0.2}{0.488} \cdot 0.64 = \frac{0.128}{0.488} = \frac{16}{61}$$

40 zgłoszeń/s $\cdot$ 86400 s $\cdot \frac{16}{61} \approx$ 907492 zgłoszenia

doba

Odp. W ciągu doby zostaje utraconych średnio 907492 zgłoszeń.

b) M|M|1|5

$$\rho = \frac{b_{śr}}{a_{śr} \cdot v} = \frac{b_{śr}}{\left(\frac{b_{śr}}{2}\right) \cdot 1} = 2$$

dwa nowe zgłoszenia w trakcie obsługi jednego

$$\rho \neq 1 \Rightarrow L = \frac{1-2}{1-2^6} \cdot 2^5 = \frac{(-1)}{(-63)} \cdot 32 = \frac{32}{63} \approx 50,8\%$$

Odp. Frakcja utraconych zgłoszeń wynosi 50,8%.

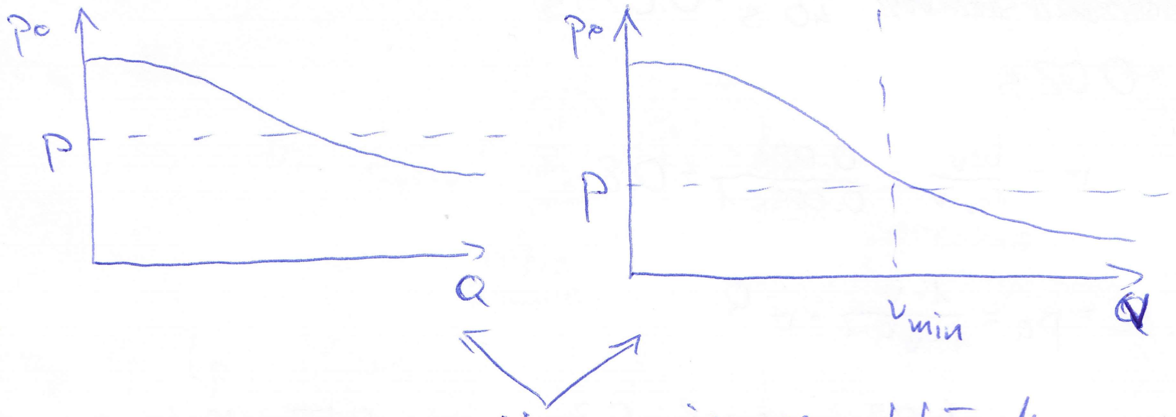
7

C)

c) M/M/1/Q

$$p_0 = \frac{1-v}{1-v^{Q+1}}, \text{ gdzie } v = \frac{b_{30}}{a_{30} \cdot v}$$

Szukamy zależności v_{min} i Q.



wykresy nie są dokładne
ilustrują tylko zależność p_0 od Q/v

Przy wzroście Q, aby zachować stałe p
v również musi rosnąć.

Solution v2: B)

5/1

~~M/M/1/2~~

b) M/M/1/5 $\quad \varrho = 5$ $\quad$ w trakcie obsługi średniego zgłoszenia przybywają średnio 2 nowe

$$2a_m = \Upsilon_m = \frac{b_m}{v} \Rightarrow \frac{b_m}{a_m v} = 2 = r$$

$L = ?$ $\quad L = \frac{1-r}{1-r^{\varrho+1}} \cdot r^{\varrho}$ $\quad$ – analiza hydrauliczna

$$L = \frac{-1}{1-2^6} \cdot 2^5 = \frac{32}{63} \approx 50\%$$

Solution v3

Jak musi wzrosnąć pamięć, aby wpo utrzymać się na 25%. ?

-6- dodatkowe 15.06.09

Zad. 10
Część danych w tabelce - nadmiarowa.

$a_{śr.}$ = średni interwał /czas zgłoszeń
V = wydajność procesu
l = frakcja strat
r = średnie obciążenie procu
p_i = frakcja czasu /prawd. gdy w systemie dana ilość zgł.
w_i = czas oczekiwania na obsługę w kolejce / czas buforowania
s = liczba procesów
d_i = czas przebywania w systemie (kol. + obsługa)
$n_{śr.}$ = średnia ilość zadań/zgłoszeń ($\in \mathbb{1}$)
$\overline{n}_{śr.}$ =
? $Q(t)$ = ilość jedn. obsługi wciąż niewykonanych

System Markowa M/M/1 - niecierpliwe I rodzaju

średni interwał między przestojem

$$\frac{p_k}{(k+1)a_{śr.}} = \frac{p_{k+1} \cdot V}{b_{śr.}} \qquad \frac{p_{k+1}}{p_k} = \frac{b_{śr.}}{V a_{śr.}(k+1)} \Rightarrow p_k = p_0 \left(\frac{b_{śr.}}{V a_{śr.}}\right)^k$$

$$p_k = \frac{r^k}{k!} \cdot p_0 \qquad \sum_{k=0}^{\infty} p_{śr.} = 1 \qquad \sum_{k=0}^{\infty} \frac{r^k}{k!} \cdot p_0 = 1 \Rightarrow$$

$$p_0 = \frac{1}{e^k}$$

C/C/1
strumień ma interwał o rozkładzie Corba

A), B)

BO Konors

dw. 5

1

a) $M/M/1/2$

liczba procesorów

pojemność pamięci (Q)

rozkład interwału między zgłoszeniami

rozkład wymagań (ile jednostek potrzebnych do wykonania zadania)

≈ 25 ms

o tyle przerwa między zgł. $= \frac{1}{a_{śr}}$

$a_{śr}$ = średnia intensywność przybywania zgłoszeń $= \frac{1}{25}$ ms

τ = średni czas przetwarzania = 20 ms = 0,02 s

L/dobę - liczba zgłoszeń utraconych w wyniku przepełnienia kolejki/bufora systemu

$r = \text{popyt}/\text{podaż} \Rightarrow \frac{\tau}{a_{śr}} = \frac{1/\tau}{a_{śr}}$

obciążenie procesora

$$L = \frac{1-r}{(1-r^{Q+1})} \cdot r^{Q}$$

dla $r \neq 1$

pojemność pamięci

$$L = \frac{1}{Q+1}$$

dla $r = 1$

frakcja utraconych zgłoszeń w wyniku przepełnienia kolejki

$r = \frac{20}{25} = \frac{4}{5}$ ms $\Rightarrow r \neq 1$, więc

$$L = \frac{1-\frac{4}{5}}{(1-\frac{4}{5}^{3})} \cdot \left(\frac{4}{5}\right)^2 = \approx 0{,}26 = 26\%$$

$40 \text{ zgł.} \cdot \underbrace{86400}_{\text{liczba sekund w dobie}} \cdot 26\% \cong 900\,000$ zgł. utraconych

b) $M/M/1/5$

$r = \frac{\tau}{a_{śr.}} \Rightarrow \frac{1}{\cancel{1/2}} \quad \frac{2 \text{ nowe}}{1 \text{ obsłużony}}$

$r \neq 1 \Rightarrow L = \frac{1-2}{(1-2^{6})} \cdot 2^5 = \frac{-1}{-63} \cdot 32 = \frac{32}{63} \approx 50\%$

C)

c) M/M/5/5 150 użytkowników (2)

~~10 trans. × 800 operacji = $\frac{1}{8000}$ s~~

~~$b_{śr.}$ - wymaganie od jedn. kolejki do obsługi zd. => 1 na operację~~

~~800 × 10 × 150 = ilość~~

⇓ $b_{śr.}$

$a_{śr.} = \frac{1}{150 \cdot 10 \cdot 800} = \frac{1}{1200000} = 0{,}83\ \mu s$

$V_1 = 0{,}5$ mln/s $\quad V_2 = 2$ mln/s

koszt = x $\quad$ koszt = $2{,}5 \cdot x$

$r = \frac{\tau}{a_{śr.}}$

obciążenie proca

~~$\tau = \frac{b_{śr.}}{V}$~~ =

$\tau_{śr.} = \underbrace{\frac{1}{500000}}_{= 2\ ms}$

$b_{śr.}$ = 800 ⇓ operacji = zgłoszenie ⇓ wymaganie (zgłoszenia) 1 operacja

$b_{śr.} = 800$

$V = 500.000$

$a_{śr} = \frac{1}{150 \cdot 10} = \frac{1}{1500} \Rightarrow r = \frac{b_{śr.}}{a_{śr.} \cdot V} = \frac{800}{1500 \cdot 500000} =$ 2,4

transakcje $S(0{,}5) = 6 \Leftarrow$ bo było $r = 2{,}4$

$S(2) = 3 \Leftarrow$ bo było $r = \frac{800}{1500 \cdot 2000000} = 0{,}6\ ms$

$6 \cdot x = 6x$

$3 \cdot 2{,}5x = 7{,}5x$ => droższe

ćw. 4 1) $\gamma = 10$ $\quad h = 4s$ $\quad b_{śr} = 15000$ $\quad V = 5000$

p_0 = wsp. bezczynności proca

$w_{śr.}$ = średnie opóźnienie buforowania (ilość czasu stracona w kolejce)

Solution v4

Zestaw V

Zadanie 1.

M/M/1/2, Q - 1 procesor, 2 miejsca na zgłoszenia

$40 \frac{zgł}{s}$, czas obsługi zgłosz. do 20ms

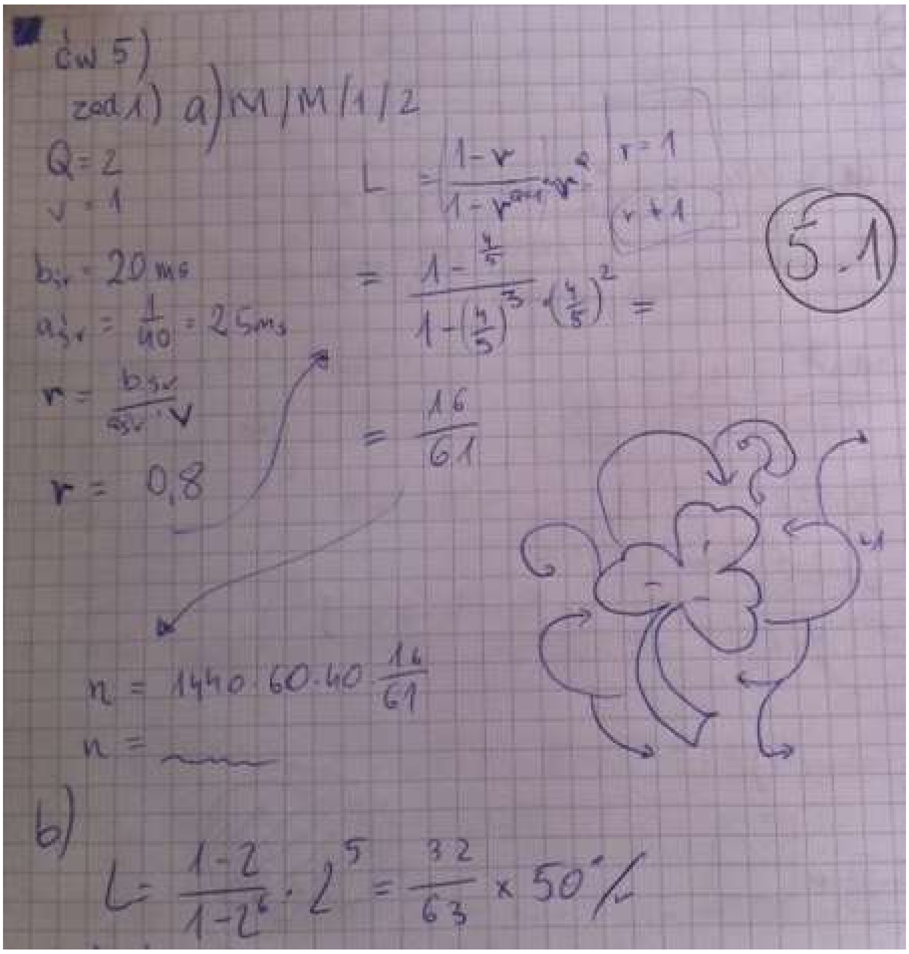
ile utraconych zgłoszeń

$a_{śr} = ? = 25ms$ (bo $40 \frac{zgł}{s}$) $\frac{1}{40 s}$ $Q = 2$

$\tau_{śr} = ? = 20 ms$

$r = \frac{\tau_{śr}}{a_{śr}}$ $r = \frac{20}{25} = 0,8$

Solution v5: A), B)



ćw 5)

zad 1) a) M/M/1/2

$Q = 2$

$v = 1$

$b_{śr} = 20 ms$

$a_{śr} = \frac{1}{40} = 25 ms$

$r = \frac{b_{śr}}{a_{śr} \cdot v}$

$r = 0,8$

$$L = \frac{1-r}{1-r^{Q+1}} \cdot r^Q \quad \Big| \; r = 1 \quad r \neq 1$$

$$= \frac{1-\frac{4}{5}}{1-\left(\frac{4}{5}\right)^3} \cdot \left(\frac{4}{5}\right)^2 = \frac{16}{61}$$

5.1

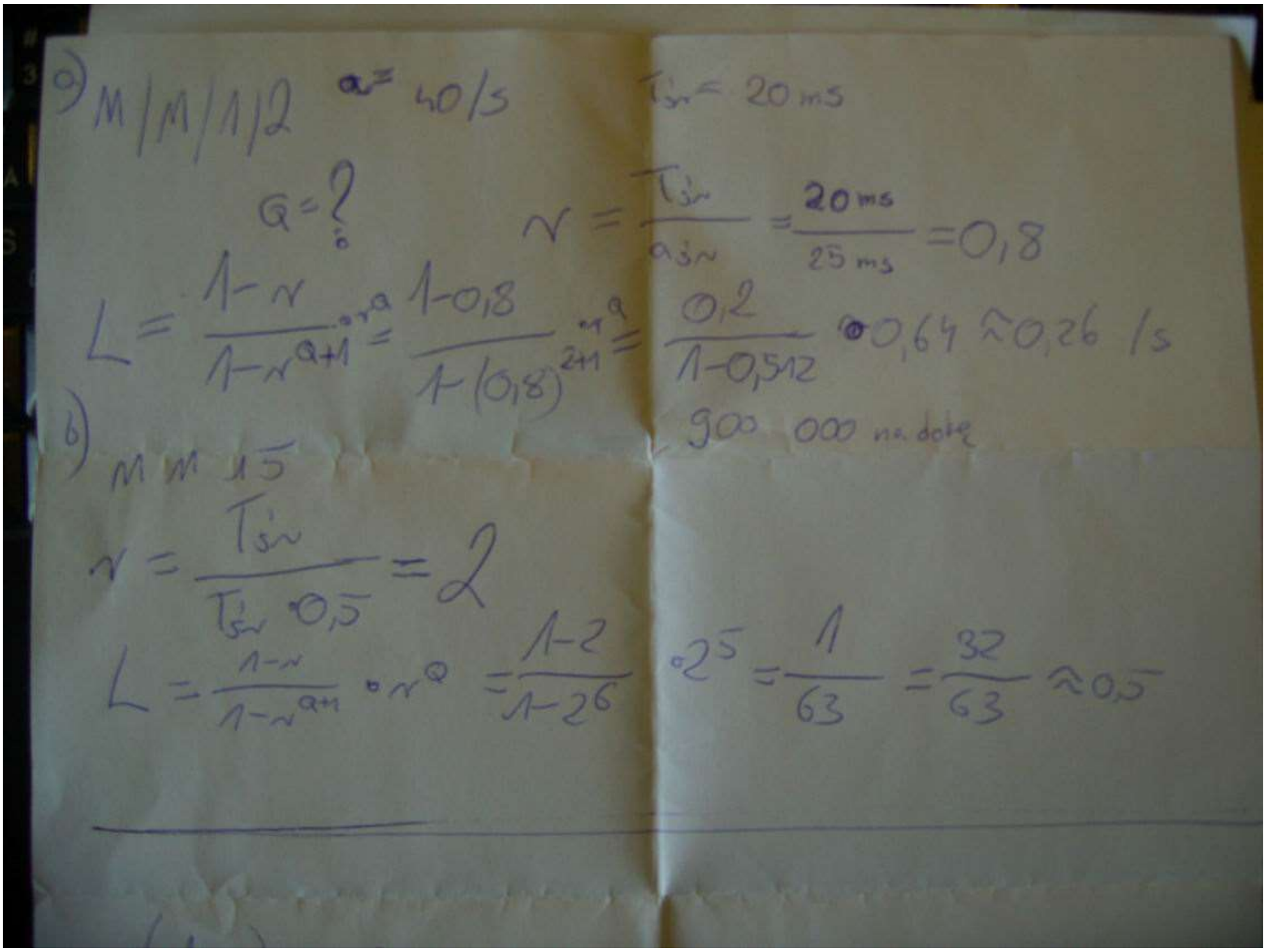
$n = 1440 \cdot 60 \cdot 40 \cdot \frac{16}{61}$

$n =$

b)

$$L = \frac{1-2}{1-2^6} \cdot 2^5 = \frac{32}{63} \times 50\%$$

Solution v5: A), B)



a) M/M/1/2 $a = 40/s$ $T_{śr} = 20\,ms$

$Q = ?$

$$r = \frac{T_{śr}}{a_{śr}} = \frac{20\,ms}{25\,ms} = 0{,}8$$

$$L = \frac{1-r}{1-r^{Q+1}} \cdot r^Q = \frac{1-0{,}8}{1-(0{,}8)^{2+1}} \cdot r^2 = \frac{0{,}2}{1-0{,}512} \cdot 0{,}64 \approx 0{,}26\ /s$$

900 000 na dobę

b) M M 1 5

$$r = \frac{T_{śr}}{T_{śr} \cdot 0{,}5} = 2$$

$$L = \frac{1-r}{1-r^{Q+1}} \cdot r^Q = \frac{1-2}{1-2^6} \cdot 2^5 = \frac{1}{63} = \frac{32}{63} \approx 0{,}5$$

Ex 5.2, Old ex

Problem

Problem 2

Consider an $M/M/S/S$ system (also called a loss system since it has no waiting room in buffer) with 150 users. Each user generates on average 10 transactions per second and tolerates $L \leq 3$ % rejections (transactions lost on arrival due to lack of available processors). An average transaction request the processor to execute 800 elementary operations. The system operator faces a choice between leasing a number of processors of speed 0.5 million elementary operations/s, each at a monthly fee, or four times faster processors leased at a 2.5 times higher monthly fee. Which type and how many processors should the operator lease to minimize the total fee while meeting the users' requirements?

In the calculations, you'll find useful the following table obtained from an Erlang-B calculator, where L values (in %) are given for $S = 1$ to 10 processors, and busy-hour load ranging from $\rho = 0.2$ to 0.6 erlangs.

S= \ $\rho=$	0.2	0.4	0.6	0.8	1	1.2	1.4	1.6	1.8	2	2.2	2.4	2.6	2.8	3
1	16.67	28.57	37.5	44.44	50	54.55	58.33	61.54	64.29	66.67	68.75	70.59	72.22	73.68	75
2	1.64	5.41	10.11	15.09	20	24.66	28.99	32.99	36.65	40	43.06	45.86	48.42	50.78	52.94
3	0.11	0.72	1.98	3.87	6.25	8.98	11.92	14.96	18.03	21.05	24	26.84	29.56	32.15	34.61
4	0.01	0.07	0.3	0.77	1.54	2.62	4	5.65	7.5	9.52	11.66	13.87	16.12	18.37	20.61
5	0	0.01	0.04	0.12	0.31	0.63	1.11	1.77	2.63	3.67	4.88	6.24	7.73	9.33	11
6	0	0	0	0.02	0.05	0.12	0.26	0.47	0.78	1.21	1.76	2.44	3.24	4.17	5.21
7	0	0	0	0	0.01	0.02	0.05	0.11	0.2	0.34	0.55	0.83	1.19	1.64	2.19
8	0	0	0	0	0	0	0.01	0.02	0.05	0.09	0.15	0.25	0.39	0.57	0.81
9	0	0	0	0	0	0	0	0	0.01	0.02	0.04	0.07	0.11	0.18	0.27
10	0	0	0	0	0	0	0	0	0	0	0.01	0.02	0.03	0.05	0.08

S= \ $\rho=$	3.2	3.4	3.6	3.8	4	4.2	4.4	4.6	4.8	5	5.2	5.4	5.6	5.8	6
1	76.19	77.27	78.26	79.17	80	80.77	81.48	82.14	82.76	83.33	83.87	84.37	84.85	85.29	85.71
2	54.94	56.78	58.48	60.07	61.54	62.91	64.19	65.39	66.51	67.57	68.56	69.49	70.38	71.21	72
3	36.95	39.15	41.24	43.21	45.07	46.83	48.49	50.06	51.55	52.97	54.3	55.57	56.78	57.92	59.02
4	22.81	24.97	27.07	29.1	31.07	32.96	34.78	36.54	38.22	39.83	41.38	42.86	44.29	45.65	46.96
5	12.74	14.51	16.31	18.11	19.91	21.68	23.44	25.16	26.84	28.49	30.09	31.64	33.15	34.62	36.04
6	6.36	7.6	8.91	10.29	11.71	13.18	14.66	16.17	17.68	19.18	20.68	22.16	23.63	25.07	26.49
7	2.83	3.56	4.38	5.29	6.27	7.33	8.44	9.6	10.81	12.05	13.32	14.6	15.9	17.2	18.5
8	1.12	1.49	1.93	2.45	3.04	3.7	4.44	5.23	6.09	7	7.97	8.97	10.01	11.09	12.19
9	0.4	0.56	0.77	1.02	1.33	1.7	2.12	2.6	3.15	3.74	4.4	5.11	5.86	6.67	7.51
10	0.13	0.19	0.28	0.39	0.53	0.71	0.93	1.18	1.49	1.84	2.24	2.68	3.18	3.72	4.31

What is the maximum tolerable busy-hour load in either of the two options?

Zadanie 2

Każdy spośród 150 użytkowników systemu M/M/*S*/*S* generuje w ciągu sekundy średnio 10 transakcji, dopuszczając $L \leq 3$ %. Każda transakcja wymaga wykonania średnio 800 operacji. Operator systemu ma do wyboru wydzierżawienie procesorów o wydajności 0.5 miliona operacji na sekundę lub procesorów 4-krotnie szybszych, których koszt dzierżawy jest 2.5-krotnie wyższy.

Ile i których procesorów powinien wydzierżawić, by zminimalizować koszty, spełniając zarazem wymagania użytkowników?

Do obliczeń wykorzystaj poniższy fragment "kalkulatora formuły B Erlanga", gdzie podano wartości L (w %) dla *normatywnych* współczynników obciążenia od 0.2 do 6 erlangów oraz dla liczb procesorów od 1 do 10.

S= \ r=	0.2	0.4	0.6	0.8	1	1.2	1.4	1.6	1.8	2	2.2	2.4	2.6	2.8	3
1	16.67	28.57	37.5	44.44	50	54.55	58.33	61.54	64.29	66.67	68.75	70.59	72.22	73.68	75
2	1.64	5.41	10.11	15.09	20	24.66	28.99	32.99	36.65	40	43.06	45.86	48.42	50.78	52.94
3	0.11	0.72	1.98	3.87	6.25	8.98	11.92	14.96	18.03	21.05	24	26.84	29.56	32.15	34.61
4	0.01	0.07	0.3	0.77	1.54	2.62	4	5.65	7.5	9.52	11.66	13.87	16.12	18.37	20.61
5	0	0.01	0.04	0.12	0.31	0.63	1.11	1.77	2.63	3.67	4.88	6.24	7.73	9.33	11
6	0	0	0	0.02	0.05	0.12	0.26	0.47	0.78	1.21	1.76	2.44	3.24	4.17	5.21
7	0	0	0	0	0.01	0.02	0.05	0.11	0.2	0.34	0.55	0.83	1.19	1.64	2.19
8	0	0	0	0	0	0	0.01	0.02	0.05	0.09	0.15	0.25	0.39	0.57	0.81
9	0	0	0	0	0	0	0	0	0.01	0.02	0.04	0.07	0.11	0.18	0.27
10	0	0	0	0	0	0	0	0	0	0	0.01	0.02	0.03	0.05	0.08

S= \ r=	3.2	3.4	3.6	3.8	4	4.2	4.4	4.6	4.8	5	5.2	5.4	5.6	5.8	6
1	76.19	77.27	78.26	79.17	80	80.77	81.48	82.14	82.76	83.33	83.87	84.37	84.85	85.29	85.71
2	54.94	56.78	58.48	60.07	61.54	62.91	64.19	65.39	66.51	67.57	68.56	69.49	70.38	71.21	72
3	36.95	39.15	41.24	43.21	45.07	46.83	48.49	50.06	51.55	52.97	54.3	55.57	56.78	57.92	59.02
4	22.81	24.97	27.07	29.1	31.07	32.96	34.78	36.54	38.22	39.83	41.38	42.86	44.29	45.65	46.96
5	12.74	14.51	16.31	18.11	19.91	21.68	23.44	25.16	26.84	28.49	30.09	31.64	33.15	34.62	36.04
6	6.36	7.6	8.91	10.29	11.71	13.18	14.66	16.17	17.68	19.18	20.68	22.16	23.63	25.07	26.49
7	2.83	3.56	4.38	5.29	6.27	7.33	8.44	9.6	10.81	12.05	13.32	14.6	15.9	17.2	18.5
8	1.12	1.49	1.93	2.45	3.04	3.7	4.44	5.23	6.09	7	7.97	8.97	10.01	11.09	12.19
9	0.4	0.56	0.77	1.02	1.33	1.7	2.12	2.6	3.15	3.74	4.4	5.11	5.86	6.67	7.51
10	0.13	0.19	0.28	0.39	0.53	0.71	0.93	1.18	1.49	1.84	2.24	2.68	3.18	3.72	4.31

Jaki będzie normatywny współczynnik obciążenia przy wolniejszych i szybszych procesorach?

Solution v1

Zadanie 2 ćwiczenia 5

M/M/S/S

$b_{30} = 800$ operacji

$$a_{30} = \frac{1}{150\text{ użytkowników} \cdot 10\text{ transakcji/s}} = \frac{1}{1500}\,s$$

$$v = \frac{b_{30}/a_{30}}{S \cdot V}, \quad S \geq 1$$

$$\rho = \frac{b_{30}}{a_{30} \cdot V}$$

procesory wolniejsze

$$v = \frac{800 \cdot \frac{1}{1500}}{500000} = 2.4$$

procesory szybsze

$$v = \frac{800 \cdot \frac{1}{1500}}{2000000} = 0.6$$

$L \leq 3\%$ (założenie)

$S_{min} = 6$ procesorów

koszt $= 6 \cdot k$

$S_{min} = 3$ procesory

koszt $= 3 \cdot 2{,}5k = 7{,}5k$

8

Solution v2

5/2 M/M/S/S - system bez oczekiwania bez blokowania

$\lambda = 150$

a_m = każdy po 10/s → 1500/s

$L \leq 3\%$

$b_m = 800$

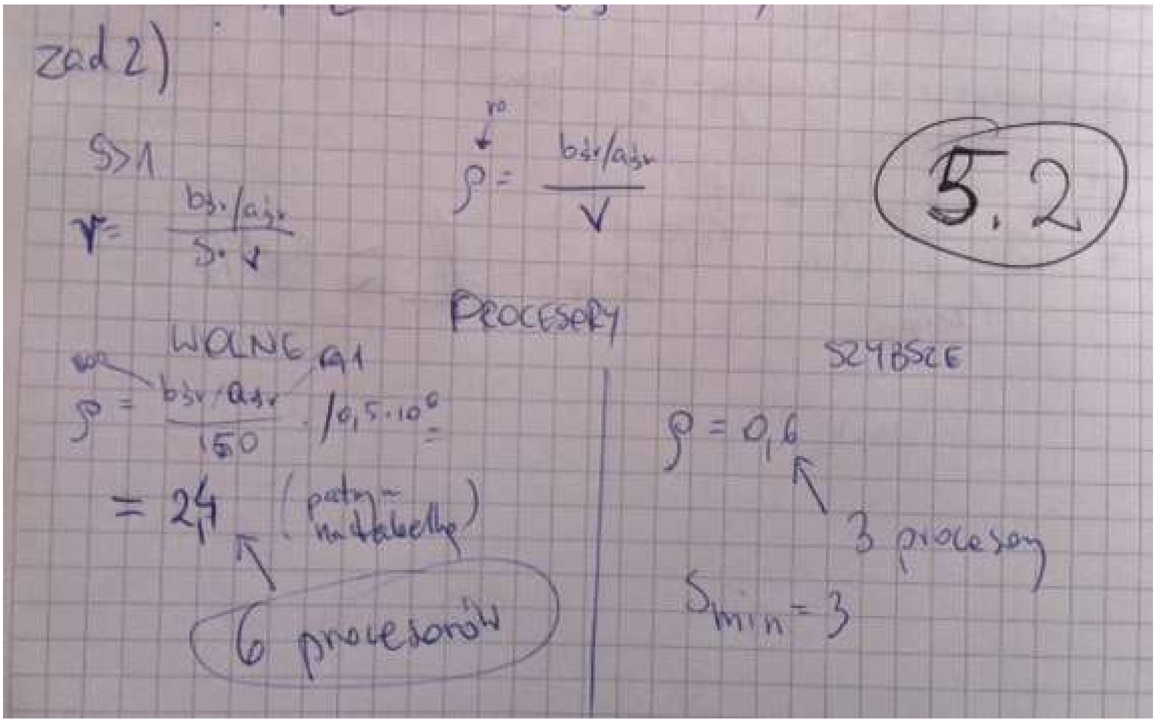
ρ - normatywny wsp. obciążenia, gdyby był 1 procesor

$$\rho = \frac{\lambda \cdot b_m}{a_m V} \leftarrow \begin{matrix} 0{,}5 \cdot 10^6 \text{ op/s} \\ \text{lub} \\ 2 \cdot 10^4 \text{ op/s} \end{matrix}$$

$\rho_1 = 2{,}4$ [erlang] → 6 cpu

$\rho_2 = 0{,}6$ [erlang] → $\geq$ 3 cpu

Solution v3



zad 2)

$S > 1$

$\rho = \frac{b_{śr}/a_{śr}}{S \cdot v}$

$\rho_0 = \frac{b_{śr}/a_{śr}}{v}$

5.2

PROCESORY

WOLNE 0,1

$\rho = \frac{b_{śr}/a_{śr}}{150} / 0{,}5 \cdot 10^6 = 2{,}4$ (patrz na tabelkę)

6 procesorów

SZYBSZE

$\rho = 0{,}6$

3 procesory

$S_{min} = 3$

Ex 5.3, Old

Problem

Problem 3

The arrival stream to an M/M/1 queuing system has mean interarrival interval $a_m = 10$ s and mean request size $b_m = 10$ s.u. Processor speed is v s.u./s. Draw a state transition diagram corresponding to the underlying birth-and-death process for the following model specifications:

a) with probability 25% a request whose service has been completed immediately returns to the queue instead of departing from the system,
b) at three or more requests in system, the processor speeds up by 50%,
c) upon termination of a busy period, the processor "goes on vacation" i.e., ignores arriving requests, and only resumes operation when three requests are queued,
d) the processor "goes on vacation" after completion of each request's service; "vacation" duration is exponentially distributed with mean h_m,
e) the processor occasionally breaks down and comes up after a while, whereupon the interrupted service is resumed (all requests arriving during breakdown are queued); the down- and up-times are exponentially distributed with mean f_m i g_m, respectively,
f) requests are admitted in pairs – the first request from a pair is held back until the second one arrives, then the pair is regarded as an arriving request of size equal to the size of the second request.

For each of these models carefully define a system state. Examine all events that can any moment occur at a given state, and the resulting state transitions.

Zadanie 3

Do systemu M/M/1 przybywa strumień zgłoszeń ze średnim interwałem $a_{śr} = 10$ s i średnim wymaganiem zgłoszenia $b_{śr} = 10$ j.o. Wydajność procesora wynosi v. Narysuj odpowiedni graf przejść stanów dla procesu urodzin i śmierci w przypadku, gdy

a) z prawdopodobieństwem 25% zgłoszenie po zakończeniu obsługi nie opuszcza systemu, lecz natychmiast powraca do kolejki,
b) przy 3 lub więcej zgłoszeniach w systemie procesor zwiększa wydajność obsługi o 50%,
c) po zakończeniu okresu zajętości procesor "idzie na wakacje", w trakcie których ignoruje zgłoszenia, zaś wznawia pracę dopiero przy 3 oczekujących zgłoszeniach,
d) każdorazowo po zakończeniu obsługi zgłoszenia procesor "idzie na wakacje", w trakcie których ignoruje znajdujące się w systemie zgłoszenia; czas trwania "wakacji" ma rozkład wykładniczy ze średnią $h_{śr}$,
e) procesor ulega awariom, w trakcie których nie obsługuje zgłoszeń (przerwana obsługa zgłoszenia zostaje dokończona po zakończeniu awarii); czasy trwania awarii oraz bezawaryjnej pracy mają rozkłady wykładnicze ze średnimi odpowiednio $f_{śr}$ i $g_{śr}$,
f) system przyjmuje zgłoszenia parami – pierwsze z pary oczekuje na następne i tak utworzona para traktowana jest jako jedno przybywające zgłoszenie o wymaganiu obsługi równym wymaganiu drugiego z pary.

Dla każdego z powyższych modeli starannie zdefiniuj stan systemu. Rozpatrz zdarzenia, jakie w każdej chwili mogą zajść w danym stanie oraz wynikające z nich stany w następnej chwili.

Zadanie 3

Do systemu M/M/1 przybywa strumień zgłoszeń ze średnim interwałem $a_{śr} = 10$ s i średnim wymaganiem zgłoszenia $b_{śr} = 10$ j.o. Wydajność procesora wynosi v. Narysuj odpowiedni graf przejść stanów dla procesu urodzin i śmierci w przypadku, gdy

a) z prawdopodobieństwem 25% zgłoszenie po zakończeniu obsługi nie opuszcza systemu, lecz natychmiast powraca do kolejki,
b) przy 3 lub więcej zgłoszeniach w systemie procesor zwiększa wydajność obsługi o 50%,
c) po zakończeniu okresu zajętości procesor "idzie na wakacje", w trakcie których ignoruje zgłoszenia, zaś wznawia pracę dopiero przy 3 oczekujących zgłoszeniach,
d) każdorazowo po zakończeniu obsługi zgłoszenia procesor "idzie na wakacje", w trakcie których ignoruje znajdujące się w systemie zgłoszenia; czas trwania "wakacji" ma rozkład wykładniczy ze średnią $h_{śr}$,
e) procesor ulega awariom, w trakcie których nie obsługuje zgłoszeń (przerwana obsługa zgłoszenia zostaje dokończona po zakończeniu awarii); czasy trwania awarii oraz bezawaryjnej pracy mają rozkłady wykładnicze ze średnimi odpowiednio $f_{śr}$ i $g_{śr}$,
f) system przyjmuje zgłoszenia parami – pierwsze z pary oczekuje na następne i tak utworzona para traktowana jest jako jedno przybywające zgłoszenie o wymaganiu obsługi równym wymaganiu drugiego z pary.

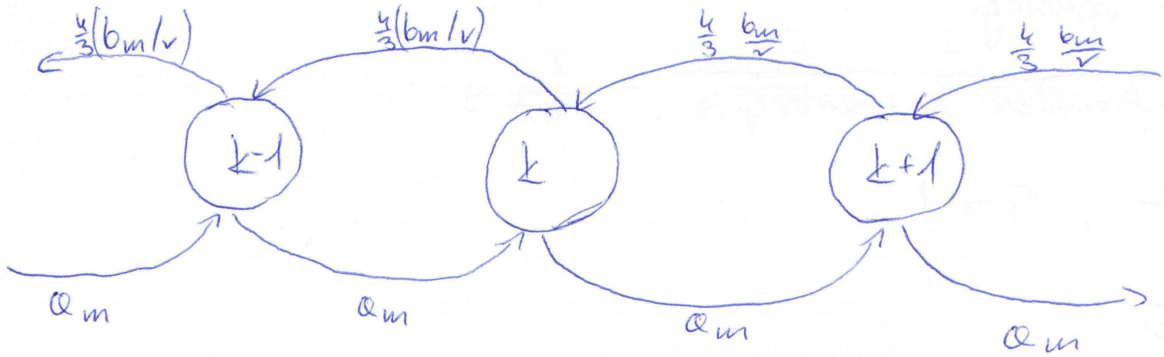
Dla każdego z powyższych modeli starannie zdefiniuj stan systemu. Rozpatrz zdarzenia, jakie w każdej chwili mogą zajść w danym stanie oraz wynikające z nich stany w następnej chwili.

Solution v1

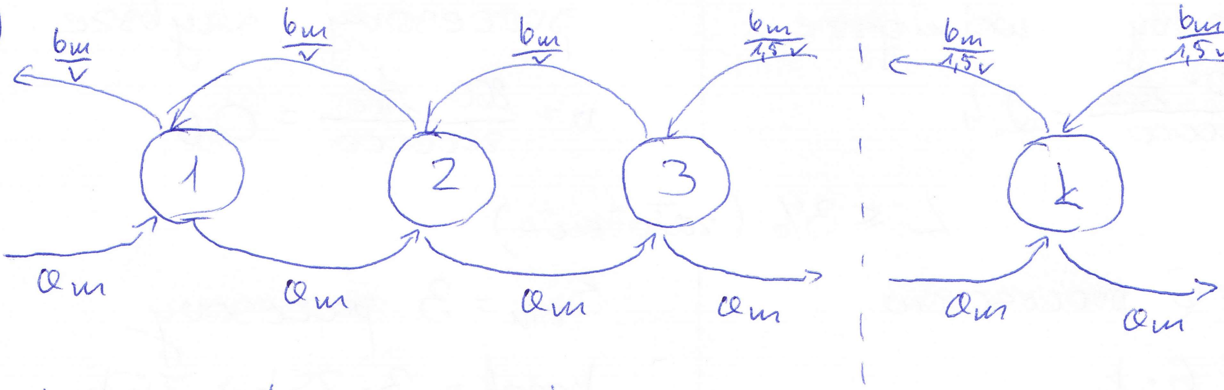
A), B), C), D)

Zadanie 3 ćwiczenia 5

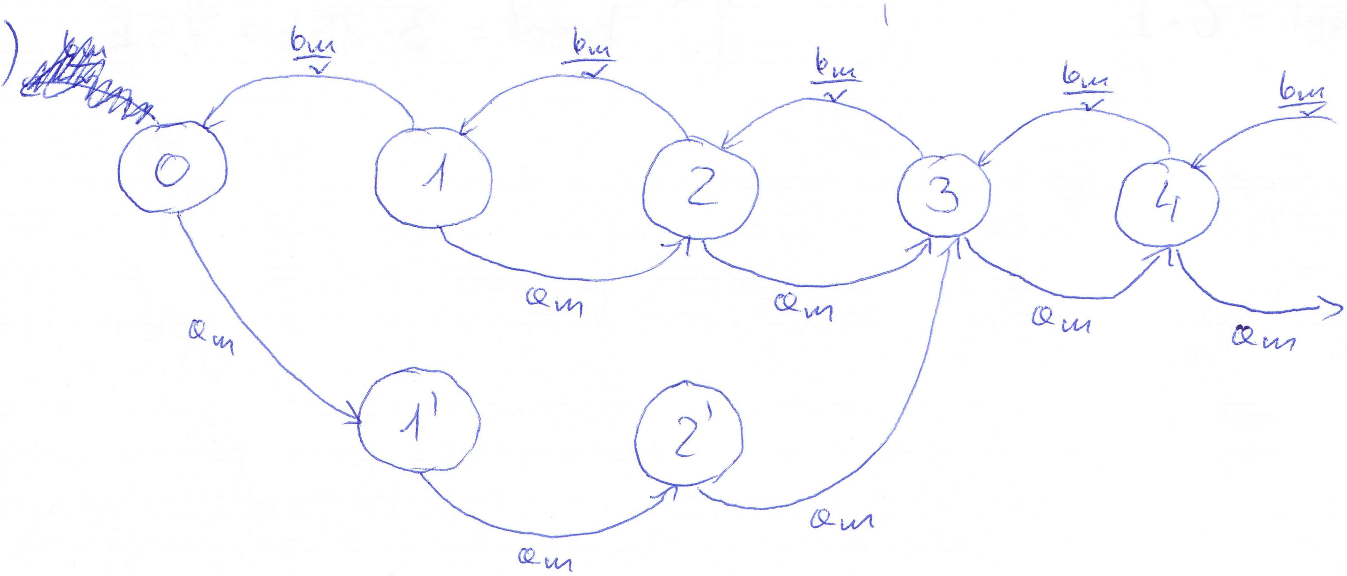
a) $v = \frac{b_{30}}{a_{30}\cdot v} = \frac{10}{(10+25\%)\cdot v} = \frac{10}{0.75v} = \frac{4}{3}v$



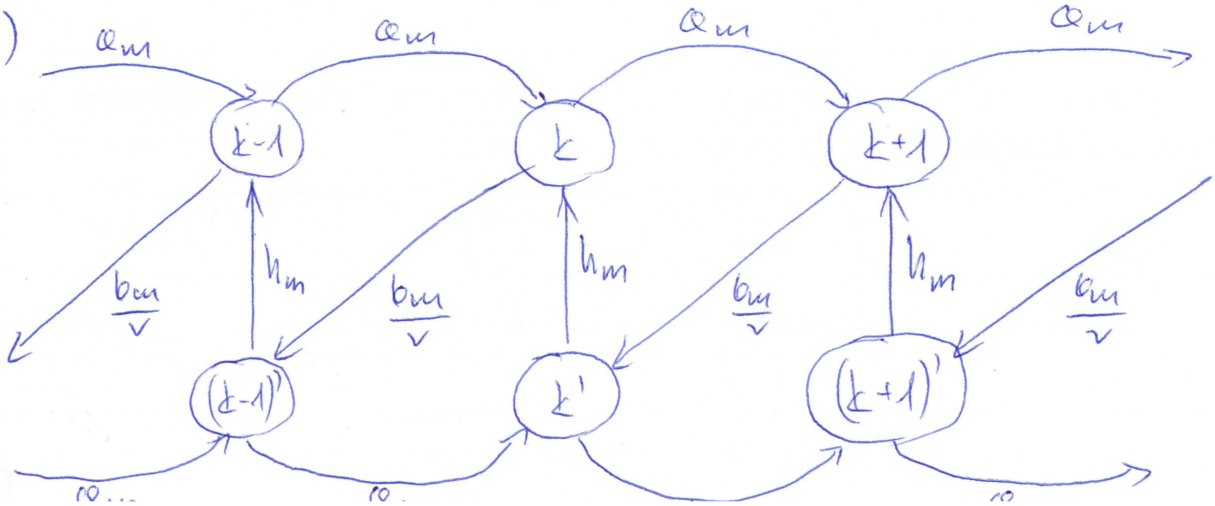
b)



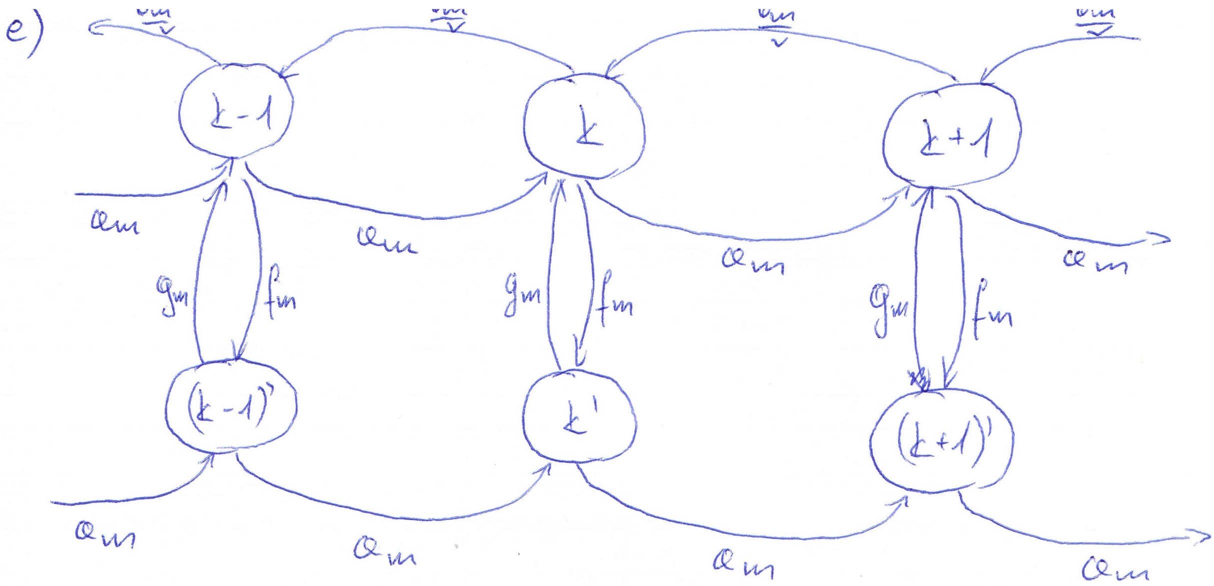
c)

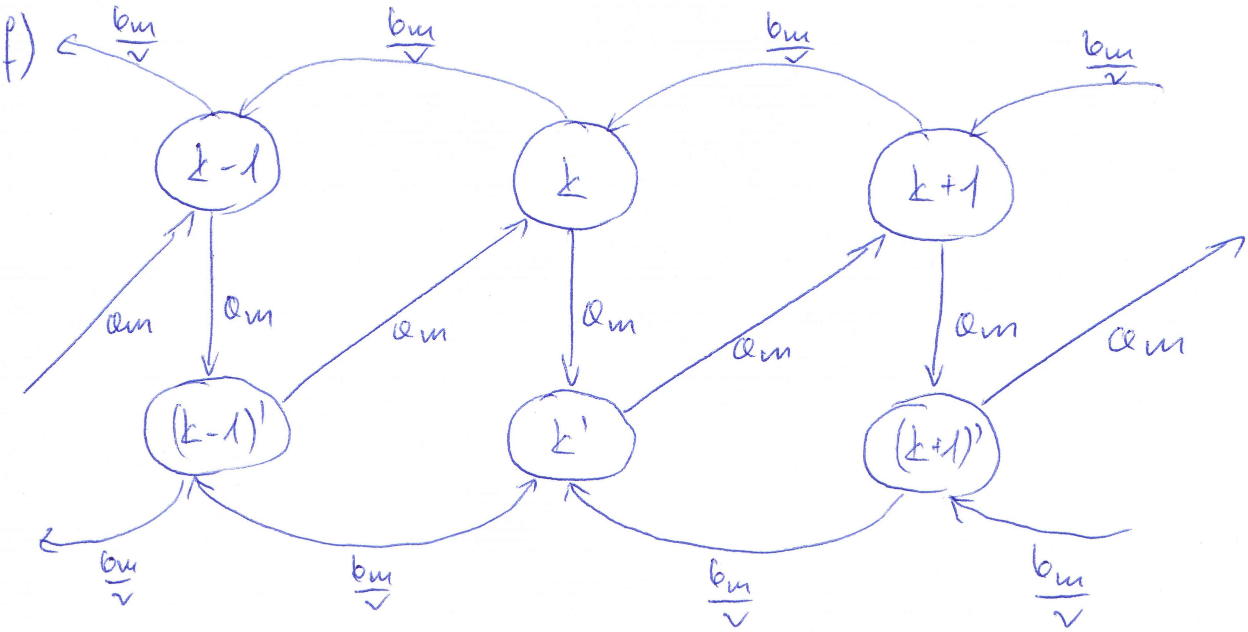


d)



E), F)



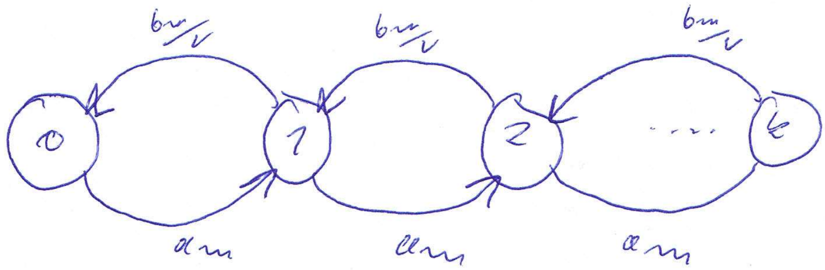


(a)

Solution v2

A), B), C)

5/3



M/M/1

a_m : 10 s

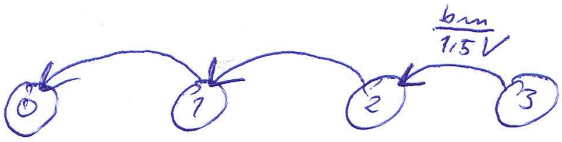
b_m = 10 op

a) 2 p = 25% zgłoszenie nie opuszcza systemu po obsłudze
75% opuszcza system

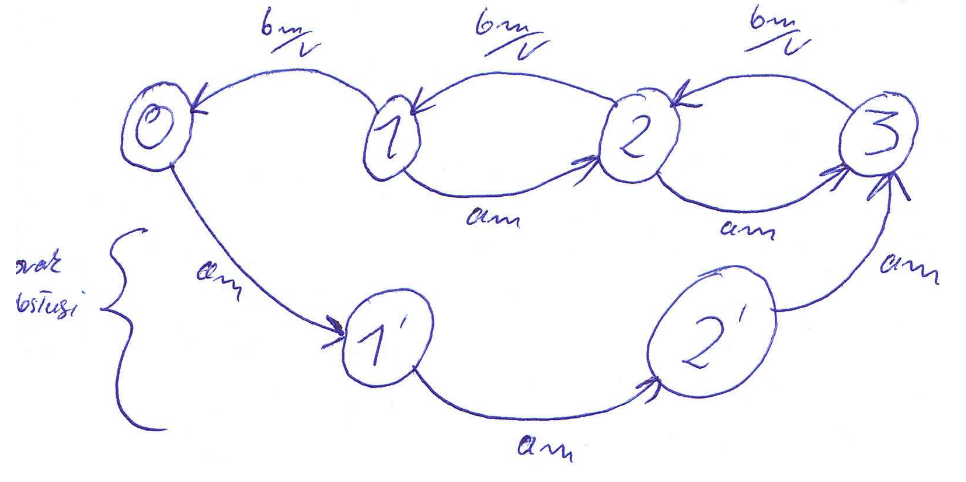
b_m wzrośnie
$b_{1m} = \frac{4}{3} b_m$

częstotliwość wyjść $= \frac{3}{4}$ częstotliwości zakończenia obsługi

b) dla 3 zgłoszeń V wzrasta do 50%
dla przypadku, gdy zgłoszeń jest powyżej 3



c) jak nie ma nic do roboty, to „idzie na wakacje”



Solution v3

A)

ĆWICZENIA 5

Zad 3

$b_{śr} = 10\mu$ $a_{śr} = 10s$ M/M/1

v

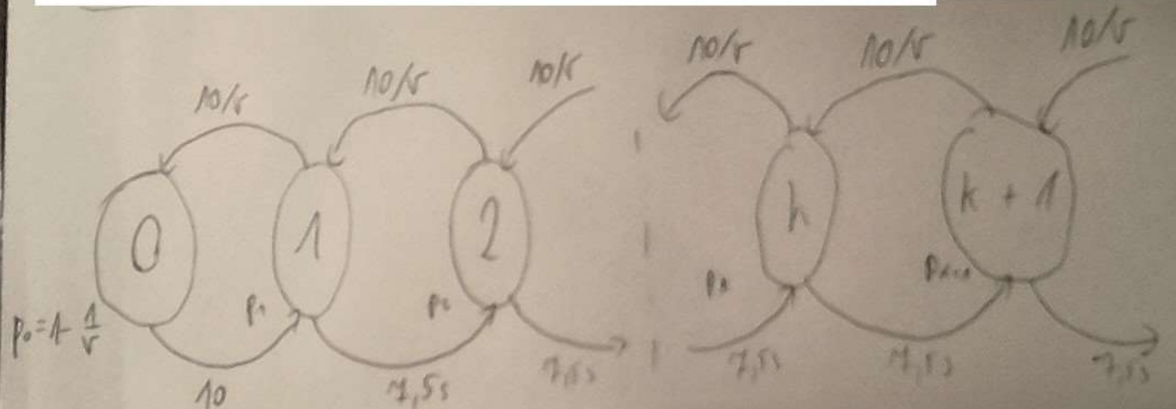
$$\boxed{r = \frac{b_{śr}/a_{śr}}{v}}$$

a) $r = \frac{1}{v}$ dla $0 \to 1$

$$r = \frac{10/7{,}5}{v} = \frac{\frac{4}{3}}{v} = \frac{4v}{3}$$

$1 - p_0 = r$

$$\boxed{p_0 = 1 - \frac{1}{v}}$$



bo 25% wcześniej

pk = p0 * r^k

$$p_0 = 1 - \frac{1}{v}$$

$$p_1 = \left(1 - \frac{1}{v}\right)\frac{1}{v}$$

$$p_2 = \left(1 - \frac{1}{v}\right)\frac{1}{v} \cdot \frac{4v}{3}$$

Na tym etapie r już jest inne bo się aśr zmniejszyło o 25%.

$$p_3 = \left(1 - \frac{1}{v}\right)\frac{1}{v}\left(\frac{4v}{3}\right)^2$$

$$p_k = \left(1 - \frac{1}{v}\right)\frac{1}{v}\left(\frac{4v}{3}\right)^{k-1}$$

$$p_{k+1} = \left(1 - \frac{1}{v}\right)\frac{1}{v}\left(\frac{4v}{3}\right)^{k}$$

Solution v4

A), B)

Practice set 5, Problem 3.

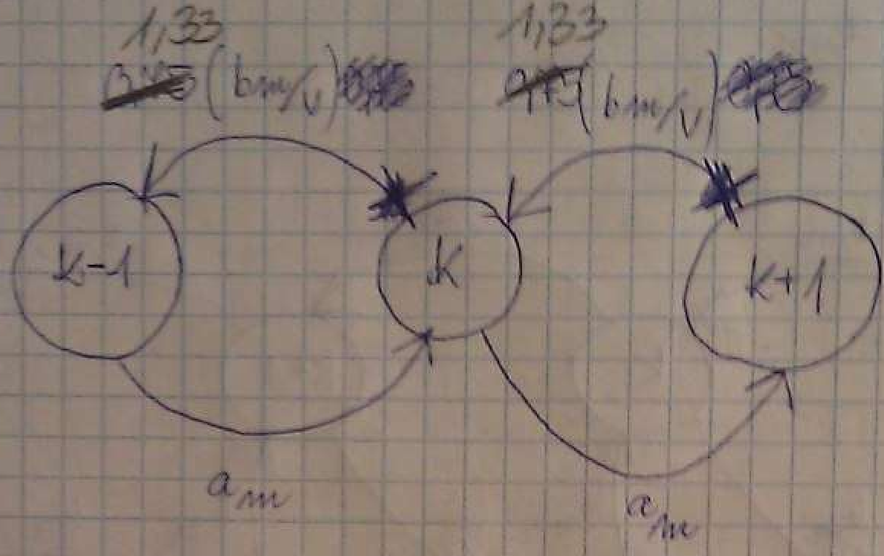
M/M/1

$a_m = 10\,s$

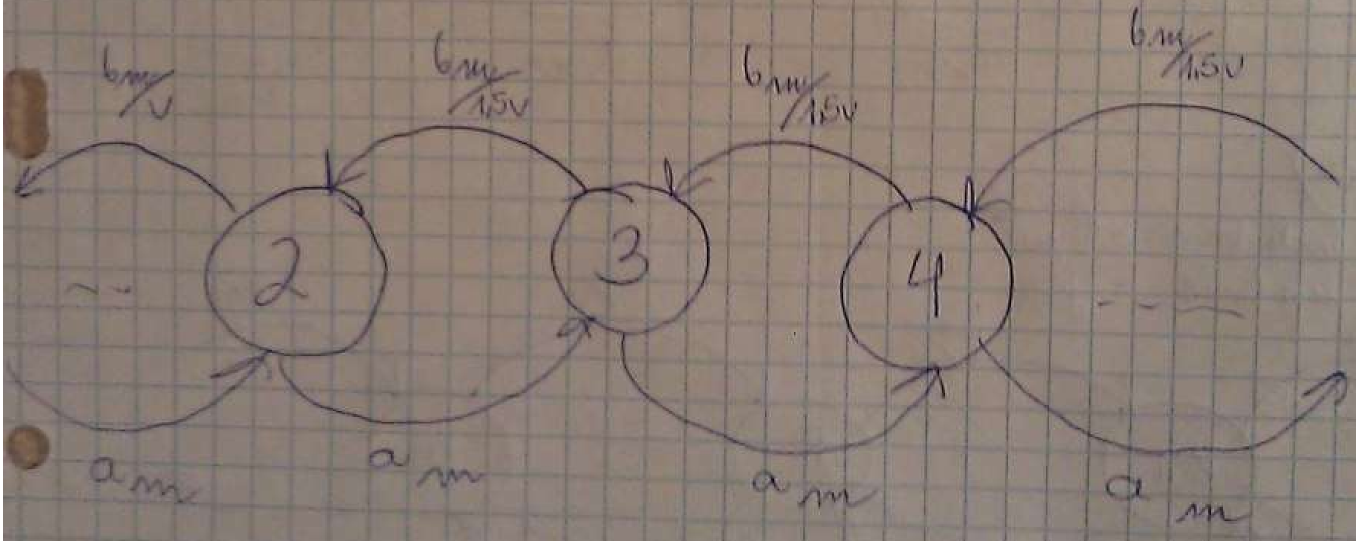
$b_m = 10$ sec

V

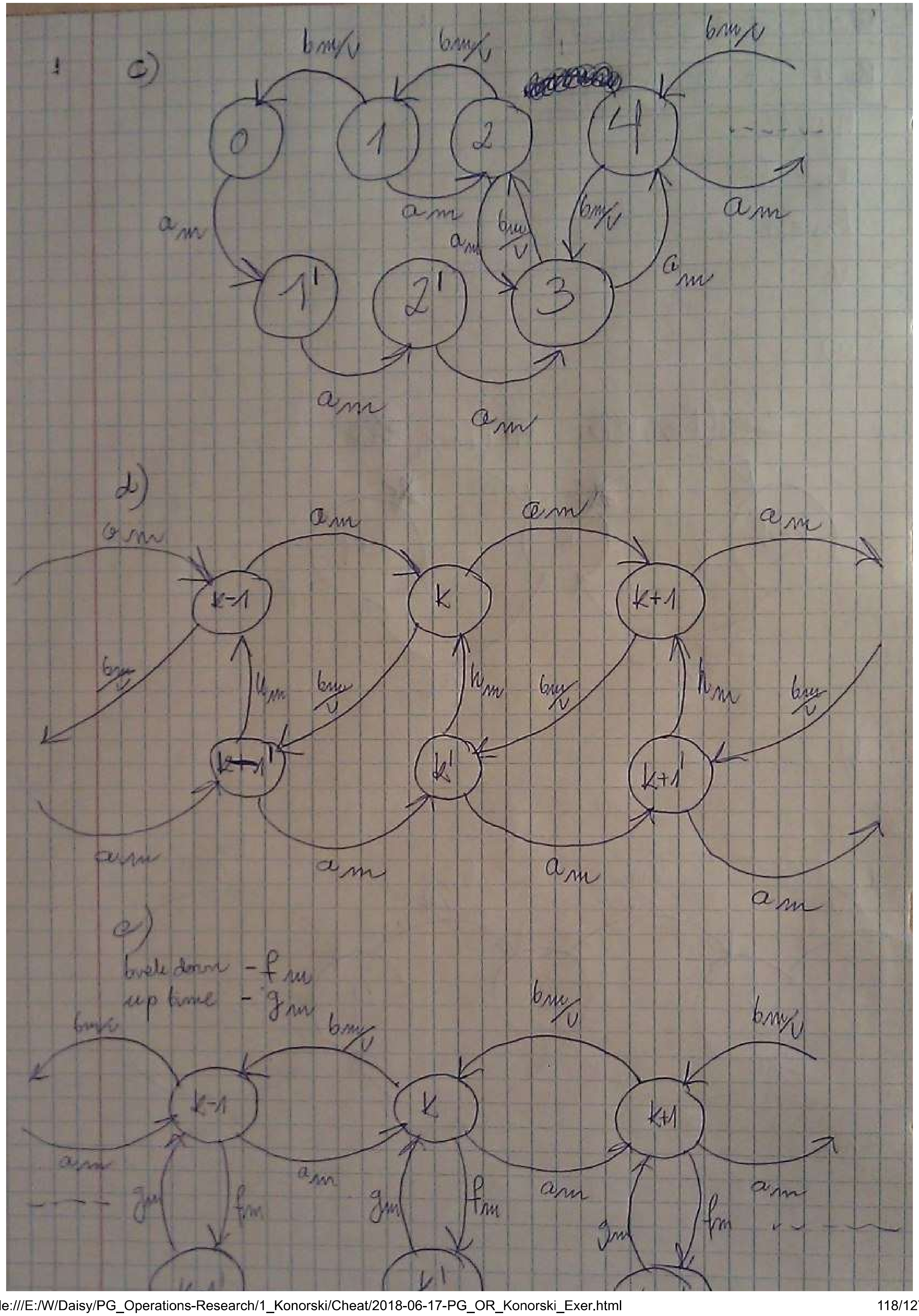
a)

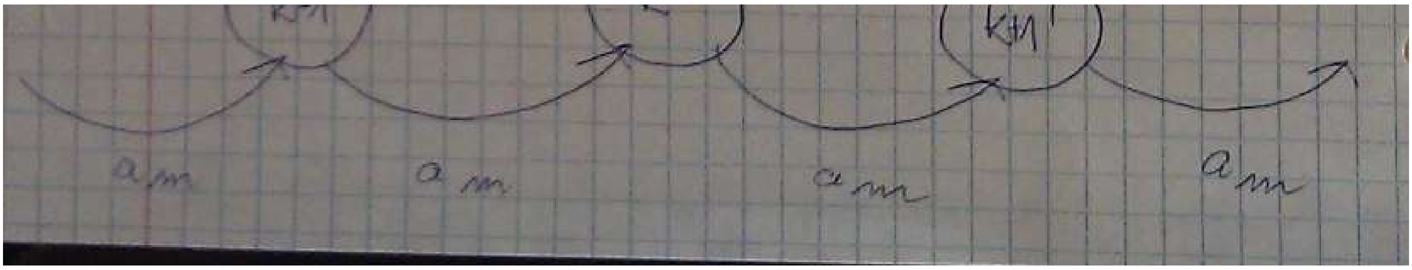


b)

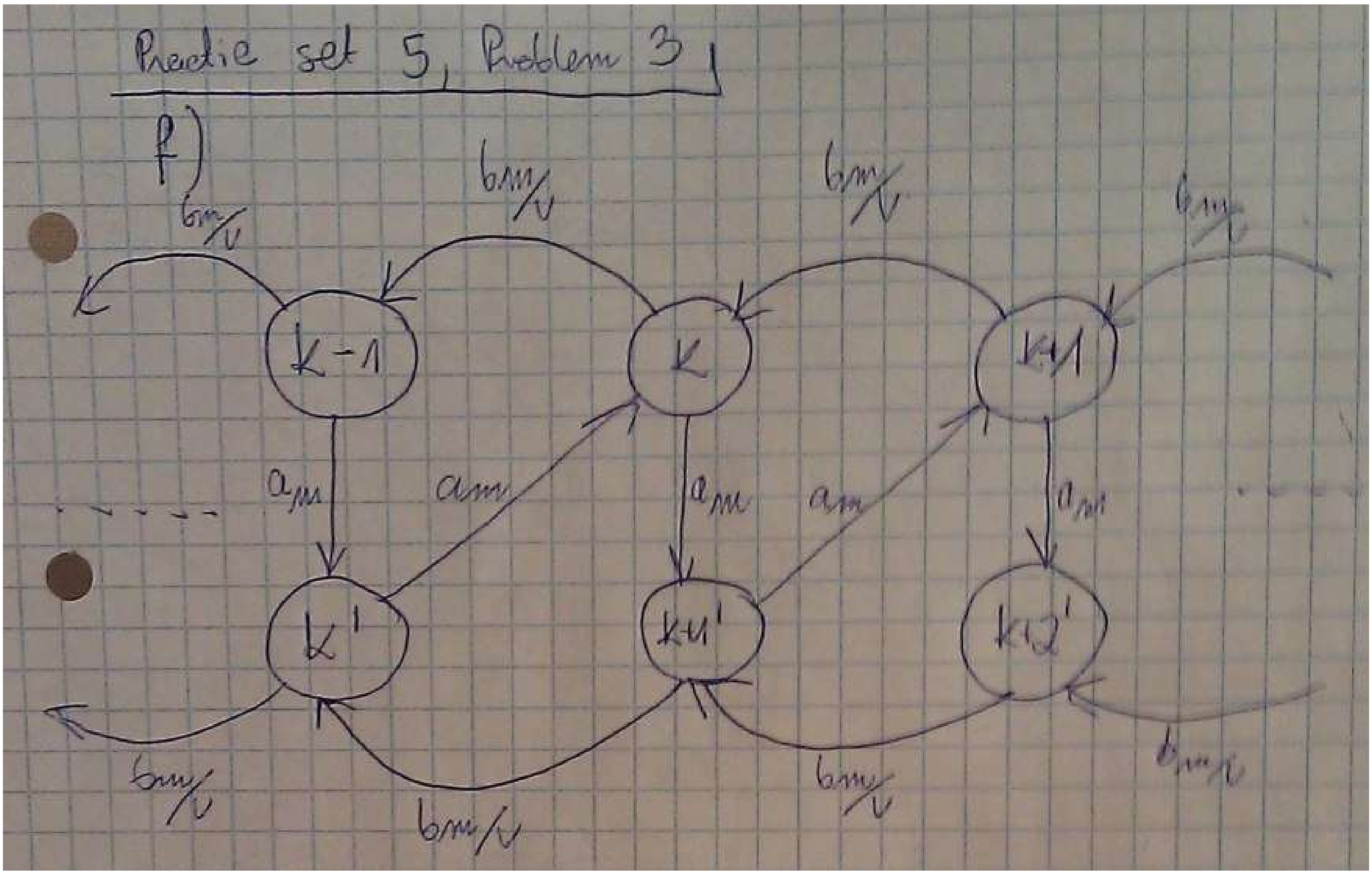


C), D), E)

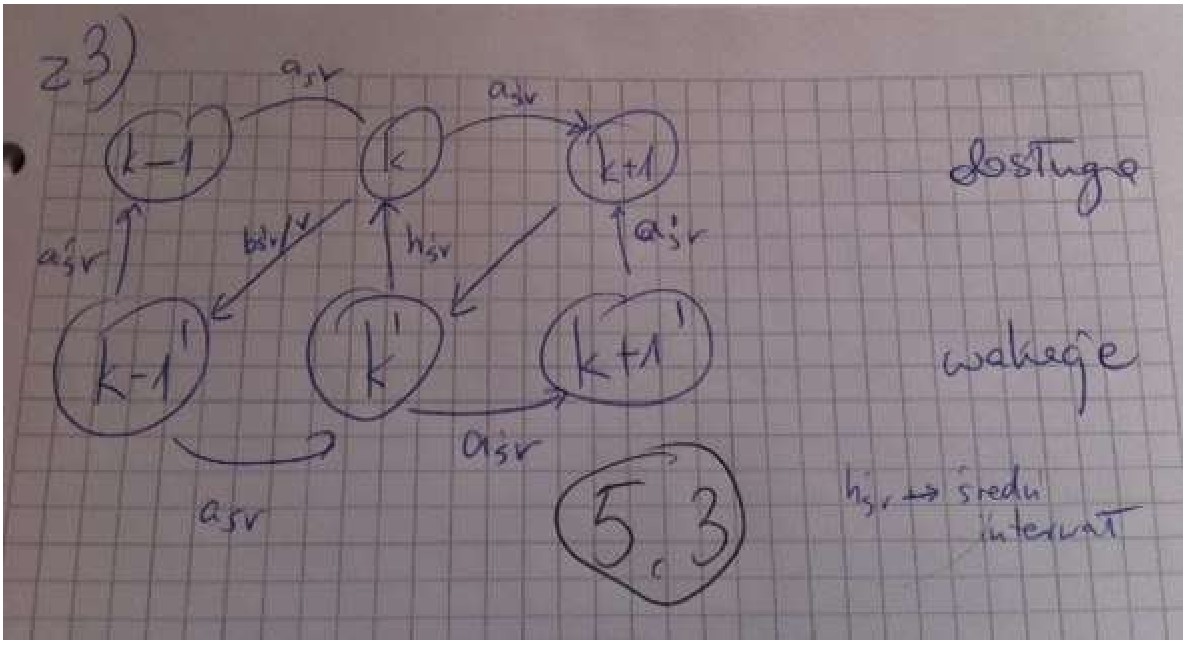

c)
bm/v
bm/v
bm/v
0
1
2
4
1'
2'
3
am
am
am
am
am
am
bm/v
bm/v
d)
am
am
am
am
k-1
k
k+1
k-1'
k'
k+1'
bm/v
bm/v
bm/v
bm/v
hm
hm
hm
am
am
am
am
e)
breakdown - fm
up time - gm
bm/v
bm/v
bm/v
bm/v
k-1
k
k+1
am
am
am
am
gm
fm
gm
fm
gm
fm
k'


k+1
k+1
a_m
a_m
a_m
a_m

F)


Practie set 5, Problem 3
f)
k-1
k
k+1
k'
k+1'
k+2'
a_m
a_m
a_m
a_m
a_m
bm/v
bm/v
bm/v
bm/v
bm/v
bm/v
bm/v
bm/v

Solution v5



Ex 5.4, Old

Problem

Problem 4

Consider impatient requests in an M/M/1 system. The arrival stream has mean interval a_m = 10 s and mean request size b_m= 10 s.u., and processor speed is v = 1 s.u./s. On arrival, each request draws its individual "patience threshold" from exponential distribution with mean c_m, and upon its expiration escapes from the queue if still waiting, otherwise departs upon service completion. For ease of calculation let $c_m = b_m/v$. Find the distribution (p_k) of the number of requests in system, as well as the fraction L of escaped requests.

Use the general birth-and-death solution, also apply the flow conservation equation.

Zadanie 4

Zgłoszenia przybywają do systemu M/M/1 średnio co $a_{śr}$ = 10 s, średnie wymaganie wynosi $b_{śr}$ = 10 j.o., zaś wydajność procesora v = 1 j.o/s. W chwili przybycia każde zgłoszenie losuje swój "próg cierpliwości" z rozkładu wykładniczego o średniej $c_{śr}$. Jeżeli po jego upływie ciągle jeszcze oczekuje na rozpoczęcie obsługi – ucieka z kolejki. Przyjmując $c_{śr} = b_{śr}/v$, znajdź rozkład liczby zgłoszeń w systemie i frakcję zgłoszeń, które uciekły z kolejki.

Posłuż się rozwiązaniem ogólnych równań urodzin i śmierci, wykorzystaj też równanie ciągłości przepływu.

Solution v1

Old ex 11 (miniprojects)

Problem

Miniprojekty:

Zadanie 11

Strumień zgłoszeń zapisany w pliku `Cw 1 zad 11.txt` (interwały w sekundach, wymagania w j.o.) należy rozdzielić na dwa systemy kolejkowe z nieograniczonymi pamięciami buforowymi i procesorami o wydajnościach v i cv tak, by średnie opóźnienia systemowe ($w_{śr} + \tau_{śr}$) w tych systemach miały się do siebie jak m : 1. W jaki sposób to osiągnąć poprzez losowe rozrzedzanie strumienia? Przyjmij dane: v = 1.25 j.o./s, c = 1.5 i m = 2. Czy to się powiedzie dla każdych v, c i m?

Solution v1

Old ex 12 (miniprojects)

Problem

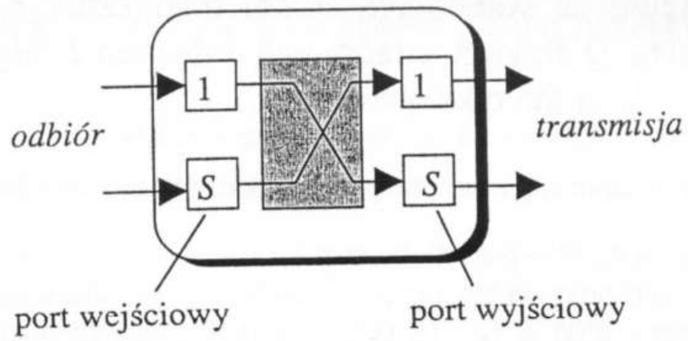
Zadanie 12

Węzeł szybkiej komutacji ramek posiada S portów wejściowych i tyle samo portów wyjściowych, pracujących z taktem równym czasowi transmisji ramki w łączu – rysunek.

Działanie portu wejściowego: O ile nie jest zajęty przez ramkę, na początku kolejnego taktu rozpoczyna odbiór następnej ramki i sprawdza, do którego portu wyjściowego jest on zaadresowany.

Działanie portu wyjściowego: tuż po początku kolejnego taktu wybiera dowolny port wejściowy zajmowany przez – lub odbierający – zaadresowaną do siebie ramkę, po czym rozpoczyna transmisję tej ramki. W braku takich portów wejściowych pozostaje bezczynny do następnego taktu.

Zakładając, że każda ramka może być z jednakowym prawdopodobieństwem zaadresowana do każdego portu wyjściowego, określ przepływność węzła (maksymalne wykorzystanie łączy wyjściowych) w zależności od S.



Solution v1

Old ex 13 (miniprojects)

Problem

Zadanie 13

Opracuj narzędzie do wykreślania przebiegu $U(t)$ jak w zadaniu 1 dla strumienia zgłoszeń zadanego w pliku oraz dowolnej liczby procesorów o dowolnych wydajnościach.

Solution v1

Old ex 14 (miniprojects)

Problem

<u>Zadanie 14</u>
Dla dyscypliny obsługi *Round Robin* jak w zadaniu 2 oraz strumienia zgłoszeń zadanego w pliku wyznacz eksperymentalnie charakterystykę czasu przebywania w systemie w funkcji wymagania zgłoszenia. Zbadaj wpływ wielkości kwantu obsługi.

Solution v1