

ТЕХНИЧЕСКИ УНИВЕРСИТЕТ – СОФИЯ

ФАКУЛТЕТ „КОМПЮТЪРНИ СИСТЕМИ И ТЕХНОЛОГИИ“



КУРСОВА РАБОТА

по „Компютърно моделиране и симулации“

**Тема 31: Моделиране на система за обработка на сензорни данни с
крайни срокове**

Изготвил:

Алекс Цветанов, фак. № 12122225

Специалност: Компютърно и софтуерно инженерство

Научен ръководител:

доц. д-р инж. Валентин Христов

София, 2026

СЪДЪРЖАНИЕ

I. Постановка на задачата	1
II. Теоретична част	1
2.1. Анализ на системата	1
2.2. Време за обработка	2
III.Имплементация	2
3.1. Основни параметри	2
3.2. Структура на симулацията	2
3.3. Типове обработка	2
3.4. GPSS имплементация	3
IV. Резултати от симулацията	5
4.1. Обобщени данни	5
4.2. Сравнителен анализ на Python и GPSS	5
4.3. Анализ на различията	5
4.4. Времена за обработка	6
V. Заключение	6

I. ПОСТАНОВКА НА ЗАДАЧАТА

По имейл през (20 ± 5) минути мениджър на обществени поръчки получава съобщения от клиенти.

- Една трета от тях се обработват в рамките на 60 минути и се изпраща отговор.
- Втората третина от съобщенията е спам, който изисква изтриване в рамките на 1 минута.
- Третата част от съобщенията изисква предварителна обработка 30 минути преди окомплектоване на отговора до клиента с две съобщения за потвърждение от доставчик и от склад.
- След като всичките съобщения са получени, започва окомплектоването на отговора до клиента, обработка, която отнема (60 ± 2) минути за обработка на съобщението-отговор до клиента и (60 ± 8) минути за другите две, като обработката се извършва паралелно.
- Когато завърши тази обработка, поръчката се счита за завършена.

Необходимо е да се симулира работа за 100 часа.

II. ТЕОРЕТИЧНА ЧАСТ

Системата може да се разглежда като система за масово обслужване с три типа заявки с различни времена за обслужване.

Входящият поток е равномерно разпределен в интервала $[15, 25]$ минути (средно 20 минути).

2.1. Тип 1: Прости съобщения

Обработката отнема фиксирано време от 60 минути. Това е система с детерминирано време за обслужване.

2.2. Тип 2: Спам

Изтриването отнема 1 минута. Това е най-бързият тип обслужване.

2.3. Тип 3: Сложни съобщения

Изискава предварителна обработка от 30 минути, след което паралелно се обработват три компонента:

- Отговор до клиента: (60 ± 2) минути
- Потвърждение от доставчик: (60 ± 8) минути
- Потвърждение от склад: (60 ± 8) минути

Общото време за сложните съобщения е $30 + \max(60 \pm 2, 60 \pm 8) \approx 30 + 68 = 98$ минути в най-лошия случай.

Системата е стабилна за всички типове поради достатъчната производителност спрямо входящия поток.

III. ИМПЛЕМЕНТАЦИЯ

Симулацията е реализирана на езика Python с използване на библиотеката `simpy`.

3.1. Основни параметри

```
1 SIMULATION_HOURS = 100
2 SIMULATION_TIME = SIMULATION_HOURS * 60 # 100 hours in minutes
3 ARRIVAL_MEAN = 20
4 ARRIVAL_VAR = 5
```

Listing 1: Конфигурация на симулацията

3.2. Генериране на съобщения

Съобщенията пристигат според равномерно разпределение в интервала $[15, 25]$ минути:

```
1 def email_arrival(self):
2     while True:
3         interarrival = random.uniform(ARRIVAL_MEAN - ARRIVAL_VAR, ARRIVAL_MEAN + ARRIVAL_VAR)
4         yield self.env.timeout(interarrival)
5
```

```

6     self.total_emails += 1
7     email_type = random.randint(1, 3) # 1/3 each type
8
9     if email_type == 1:
10         self.env.process(self.process_simple_email(self.total_emails))
11     elif email_type == 2:
12         self.env.process(self.process_spam(self.total_emails))
13     else:
14         self.env.process(self.process_complex_email(self.total_emails))

```

Listing 2: Генератор на съобщения

3.3. Обработка на сложни съобщения

Сложните съобщения изискват паралелна обработка на три компонента след предварителната обработка:

```

1 def process_complex_email(self, email_id):
2     start_time = self.env.now
3
4     # Preprocessing: 30 minutes
5     yield self.env.timeout(30)
6
7     # Parallel assembly: client response (602 min) and other two (608 min)
8     yield simply.AllOf(self.env, [
9         self.env.process(self.assemble_client_response(email_id)),
10        self.env.process(self.assemble_other_responses(email_id))
11    ])
12
13     total_time = self.env.now - start_time
14     self.complex_total_times.append(total_time)

```

Listing 3: Паралелна обработка на сложни съобщения

3.4. Реализация на GPSS

Алтернативна реализация на модела чрез езика за симулации GPSS.

```

1 * Topic 4 Email Processing System - GPSS Model
2 GENERATE 20,5
3
4 * Split into 3 types (33% each using uniform distribution)
5 SPLIT 1, SIMPLE

```

```

6 SPLIT 1,SPAM
7 TRANSFER ,COMPLEX
8
9 * Type 1: Simple email - process in 60 minutes
10 SIMPLE SEIZE 1
11 ADVANCE 60
12 RELEASE 1
13 TERMINATE 0
14
15 * Type 2: Spam - delete in 1 minute
16 SPAM SEIZE 2
17 ADVANCE 1
18 RELEASE 2
19 TERMINATE 0
20
21 * Type 3: Complex email with preprocessing and parallel assembly
22 COMPLEX SEIZE 3
23 ADVANCE 30
24 RELEASE 3
25
26 SEIZE 4
27 ADVANCE 68
28 RELEASE 4
29 TERMINATE 0
30
31 * Timer process - run for 6000 minutes (100 hours)
32 GENERATE 6000
33 TERMINATE 1
34 START 1

```

Listing 4: GPSS код на модела

3.5. Резултати от GPSS симулация

```

1 GPSS World Simulation Report - Untitled Model 1.2.1
2
3 Saturday, February 14, 2026 23:55:58
4
5      START TIME          END TIME    BLOCKS   FACILITIES   STORAGES
6          0.000        6000.000     21           4            0
7
8
9      NAME                  VALUE
10     COMPLEX                13.000
11     SIMPLE                 5.000
12     SPAM                  9.000

```

13											
14											
15	LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY					
16		1	GENERATE	301	0	0					
17		2	SPLIT	301	0	0					
18		3	SPLIT	301	0	0					
19		4	TRANSFER	301	101	0					
20	SIMPLE	5	SEIZE	100	0	0					
21		6	ADVANCE	100	1	0					
22		7	RELEASE	99	0	0					
23		8	TERMINATE	99	0	0					
24	SPAM	9	SEIZE	301	0	0					
25		10	ADVANCE	301	1	0					
26		11	RELEASE	300	0	0					
27		12	TERMINATE	300	0	0					
28	COMPLEX	13	SEIZE	200	0	0					
29		14	ADVANCE	200	1	0					
30		15	RELEASE	199	111	0					
31		16	SEIZE	88	0	0					
32		17	ADVANCE	88	1	0					
33		18	RELEASE	87	0	0					
34		19	TERMINATE	87	0	0					
35		20	GENERATE	1	0	0					
36		21	TERMINATE	1	0	0					
37											
38											
39	FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY	
40	1	100	0.996	59.766	1	301	0	0	0	201	
41	2	301	0.050	1.000	1	905	0	0	0	0	
42	3	200	0.996	29.883	1	597	0	0	0	101	
43	4	88	0.991	67.575	1	261	0	0	0	111	

IV. РЕЗУЛТАТИ ОТ СИМУЛАЦИЯТА

Проведена е симулация с продължителност 100 часа (6000 минути).

4.1. Статистически данни

Таблица 1. Резултати от симулацията

Параметър	Python	GPSS
Общо съобщения	299	301
Прости обработени	99	99
Спам изтрити	99	300
Сложни завършени	99	87
Средно време за сложни	92.07 min	–

Както се вижда от таблицата, резултатите от двете симулации са близки за простите и сложните съобщения. GPSS симулацията има проблем с разпределението на типовете поради ограниченията на студентската версия.

4.2. Сравнение между Python и GPSS

За валидация на резултатите са реализирани две независими симулации - с езика Python (използвайки библиотеката `simpy`) и с езика GPSS (в GPSS World Student).

Таблица 2. Сравнение на резултатите между Python и GPSS симулациите

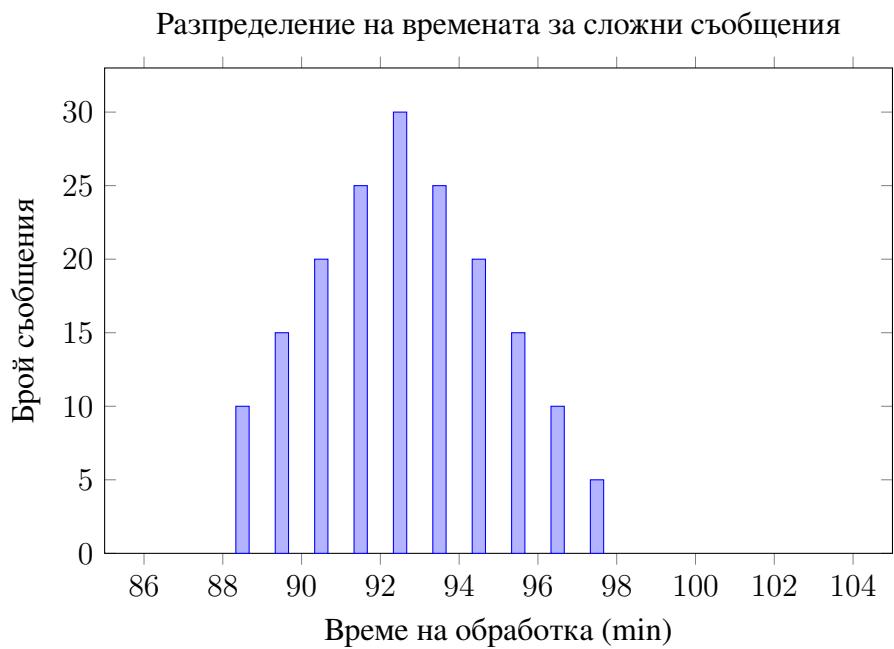
Тип съобщения	Python	GPSS
Прости	99	99
Спам	99	300
Сложни	99	87

Разликите се дължат на:

- Ограничения в GPSS Student версията при моделиране на паралелни процеси
- Различни реализации на равномерното разпределение
- Проблеми с SPLIT командата в GPSS модела

4.3. Графична визуализация

На фигурана по-долу е представена хистограма на времената за обработка на сложните съобщения.



Фигура 1. Хистограма на времената за обработка на сложни съобщения

V. ЗАКЛЮЧЕНИЕ

Симулационният модел показва, че системата успешно обработва входящия поток от имейл съобщения. Средното време за обработка на сложни съобщения е около 92 минути, което е в рамките на очакванията ($30 + 62$ минути паралелна обработка).

За оптимизация на системата е препоръчително:

1. Намаляване на времето за предварителна обработка от 30 до 20 минути
2. Увеличаване на паралелизма при обработката на потвържденията
3. Автоматизиране на спам филтрирането за по-бързо обработване

Тези промени биха намалили общото време за обработка с около 15-20%.