به نام خدا

اصول شبیه سازی

گزارش از روند حل مسائل

استاد درس

جناب آقای دکتر احمدی

تهیه کنندگان

فاطمه نیک خواه - ۸۸۲۸۰۹۶

علی لشگری - ۸۷۵۸۷۱۷

محمدرضا تأثیری - ۸۸۱۲۰۷۶

[This page intentionally left blank]

Fork me on GitHub

Fork Source Code on Github

https://github.com/taesiri/Simulation

We do love Open Source, Full Source code and project documentation is available on github.com/taesiri/Simulation.

Taesiri: The source code is published under my Github account just because I was the only person had active Github account at publishing date.

فهرست مطالب

A Brief Technical Document on Projects Implementation

مختصر اطلاعات فنی در مورد پیاده سازی پروژها

Total Line of Codes: 1349 (Problem 1 with 467 and Problem 2 with 882) – Calculated with Code Metrics

Programming Languages Used: C#, IronPython, IronRuby

Application User interface entirely written in WPF using C#

Runtime Platform: .Net Framework 4 (Scripting features needs host has IronPython and/or IronRuby Installed)

زبان برنامه نویسی اصلی، زبان C# می باشد. تعداد خطوط نوشته شده در حدود ۱۵۰۰ خط برای هر دو مساله می باشد.

مساله ۱

مساله پایایی مربوط به ماشین فرز دارای ۳ برینگ را برای هر دو روش پیشنهادی؛ برای ۲۰ خرابی یا تعویض از هر برینگ شبیه سازی نمایید. نمودارهای جریان مربوطه را نیز برای هر دو روش ترسیم نمایید.

(سوال اشاره ای به مقایسه دو روش نکرده)

(در شبیه سازی این مساله، کارگر برای تعمیر دستگاه به تعداد کافی در اختیار داریم و صفی تشکیل نمی شود)

مقایسه دو روش

با تکرار آزمایش به تعداد زیاد، روش دوم را روش بهینه معرفی می کنیم. مشاهده شده است که زمان خواب دستگاه کاسته شده است.

Conclusion

	Cumulative Life time	Wasted Time
Bearing 1	32500	86
Bearing 2	28800	94
Bearing 3	28000	102

روش اول

Conclusion

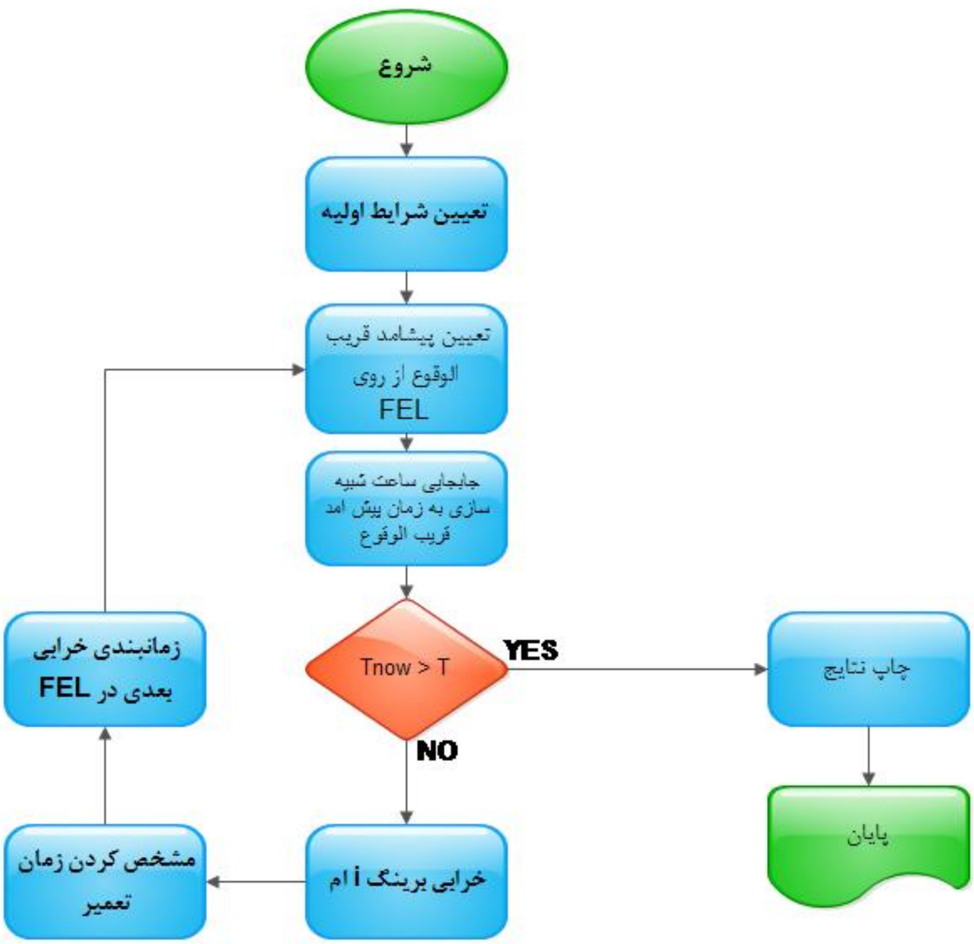
	Cumulative Life time	Wasted Time
Total	25300	93

روش دوم

برای مشاهده نمونه گزارش های شبیه سازی به ضمیمه گزارش مراجعه کنید.

Flow Chart

تذکر : در شبیه سازی این مساله، برای بهبود سرعت (Performance) از روش FEL استفاده نشده؛ گرچه نمودارهای جریان بر اساس FEL گزارش داده شده اند.



Bearings - Flow Chart

مساله ۲

Special thanks to Google Goggle!

Able, Baker, and Charlie are three carhops at the Sonic Drive-In (service at the speed of Sound). Cars arrive every 5 +- 5 minutes. The carhops service customers at the rate of one 10+-6 minutes. However the customers prefer Able over Baker and Baker over Charlie. If the carhop of choice is busy, the customers choose the first available carhop.

Simulate the system for 1000 service completions. Estimate Able's, Baker's and Charlie's Utilization (percentage of time busy).

Also, repeat the case where there are no preferences between carhops. Compare the results with the previous case. Which approach would be better?

Source:

Exercise problems - Ugrad.cs.ubc.ca

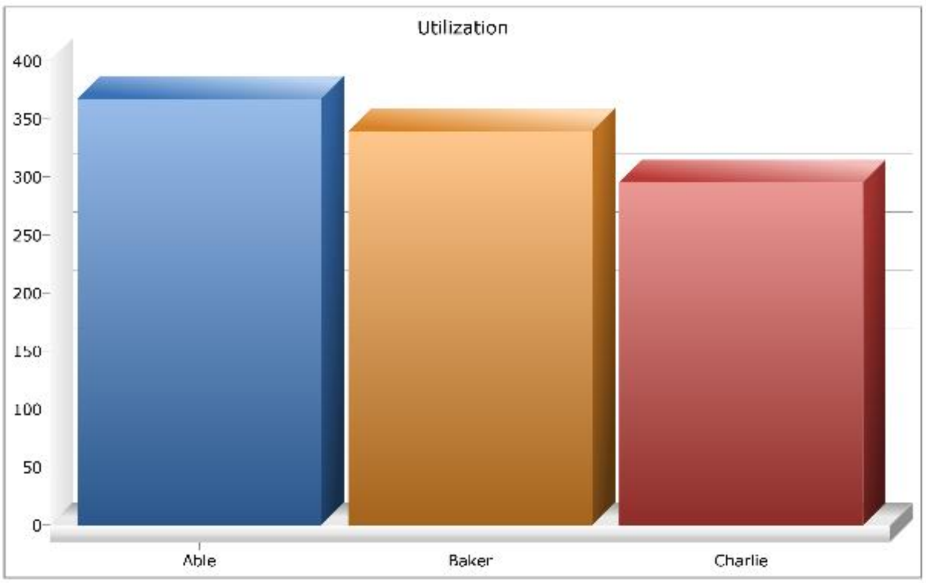
www.ugrad.cs.ubc.ca/~cs405/ch4excercises.pdf

مقایسه دو روش

با تکرار آزمایش به تعداد زیاد، در روش اول، Able بیشترین زمان خدمت دهی و بیشترین تعداد سرویس به مشتری را به خود اختصاص داده است. بعد از او Baker و Charlie به ترتیب قرار می گیرند.

نمونه ای از خروجی برنامه در حالت اول

Utilization Charts



Generated Report

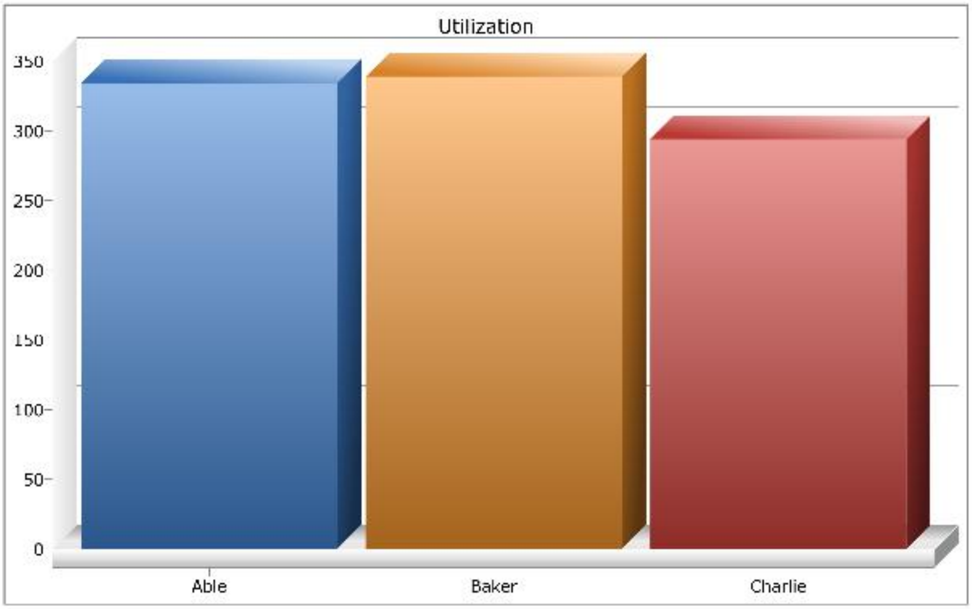
Solving case with Preferences between carhops

No.	Arrival Time	r. Number	Service Time	Queue Time	Service Provider	Departure Time	State *(On Arrival)*	State *(After Departure)*
1	0	0	7	0	Able	7	(0,0,0)	(0,1,1)
2	3	3	10	0	Baker	13	(1,0,0)	(1,0,1)
3	5	2	11	0	Charlie	16	(1,1,0)	(1,1,0)
4	6	1	16	1	Able	23	(1,1,1)	(0,1,1)
5	10	4	4	3	Baker	17	(1,1,1)	(1,0,1)
6	10	0	5	6	Charlie	21	(1,1,1)	(1,1,0)
7	13	3	8	4	Baker	25	(1,1,1)	(0,0,1)
8	21	8	10	0	Charlie	31	(1,1,1)	(1,0,0)
9	30	9	12	0	Able	42	(0,0,1)	(0,1,0)
10	39	9	9	0	Baker	48	(1,0,0)	(1,0,0)

در شبیه سازی به روش دوم مشاهده می شود این میزان مشارکت سه خدمت دهنده به طور متوسط یکی است

نمونه ای از خروجی در حالت دوم

Utilization Charts



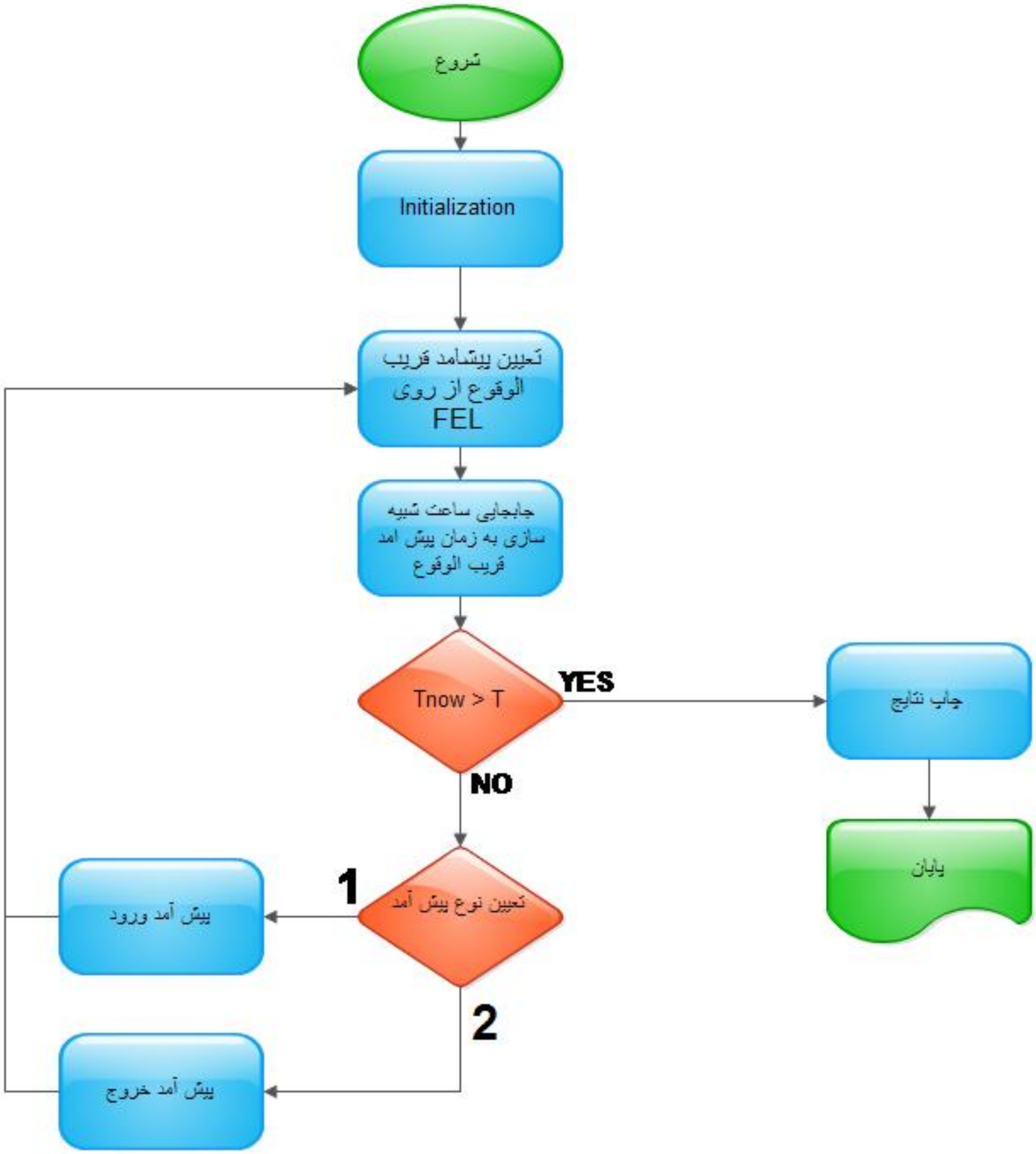
Generated Report

Solving case with no Preferences between carhops

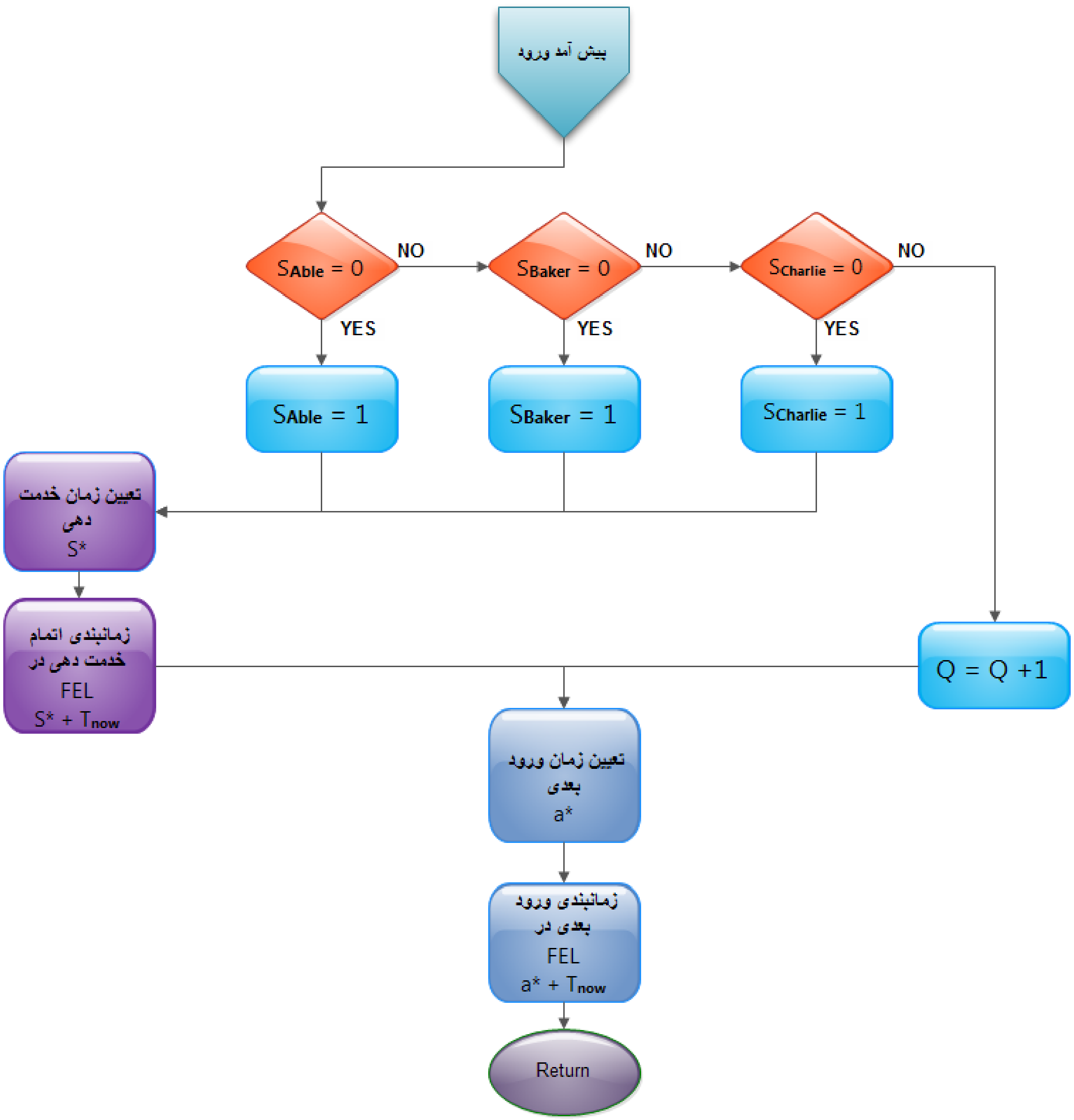
No.	Arrival Time	r. Number	Service Time	Queue Time	Service Provider	Departure Time	State *(On Arrival)*	State *(After Departure)*
1	0	0	7	0	Baker	7	(0,0,0)	(1,0,1)
2	4	4	7	0	Able	11	(0,1,0)	(0,1,1)
3	6	2	5	0	Charlie	11	(1,1,0)	(1,1,0)
4	8	2	4	0	Baker	12	(1,0,1)	(1,0,0)
5	8	0	10	3	Able	21	(1,1,1)	(0,1,1)
6	13	5	13	0	Baker	26	(1,0,0)	(1,0,1)

Class Diagram

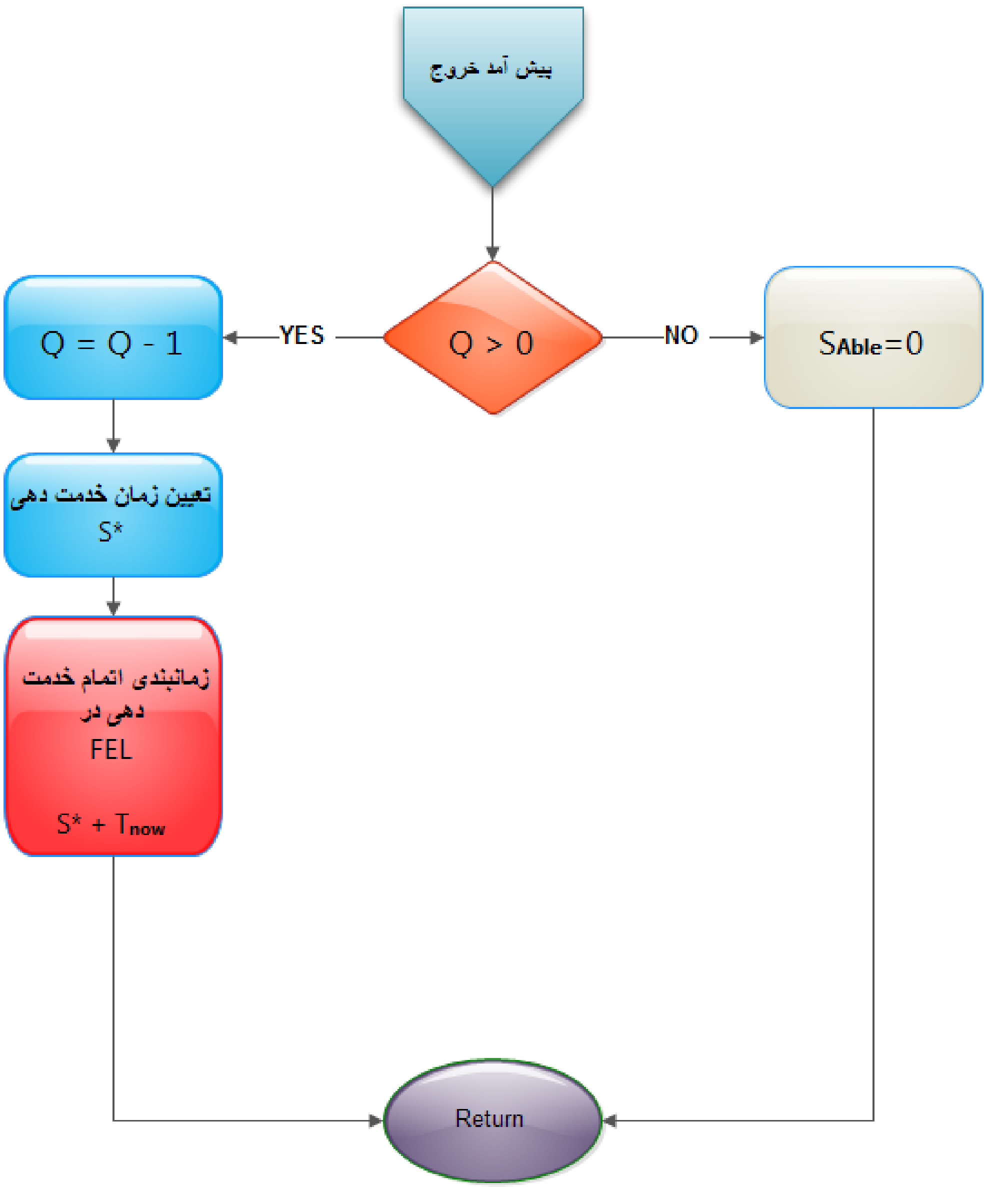
Flow Charts



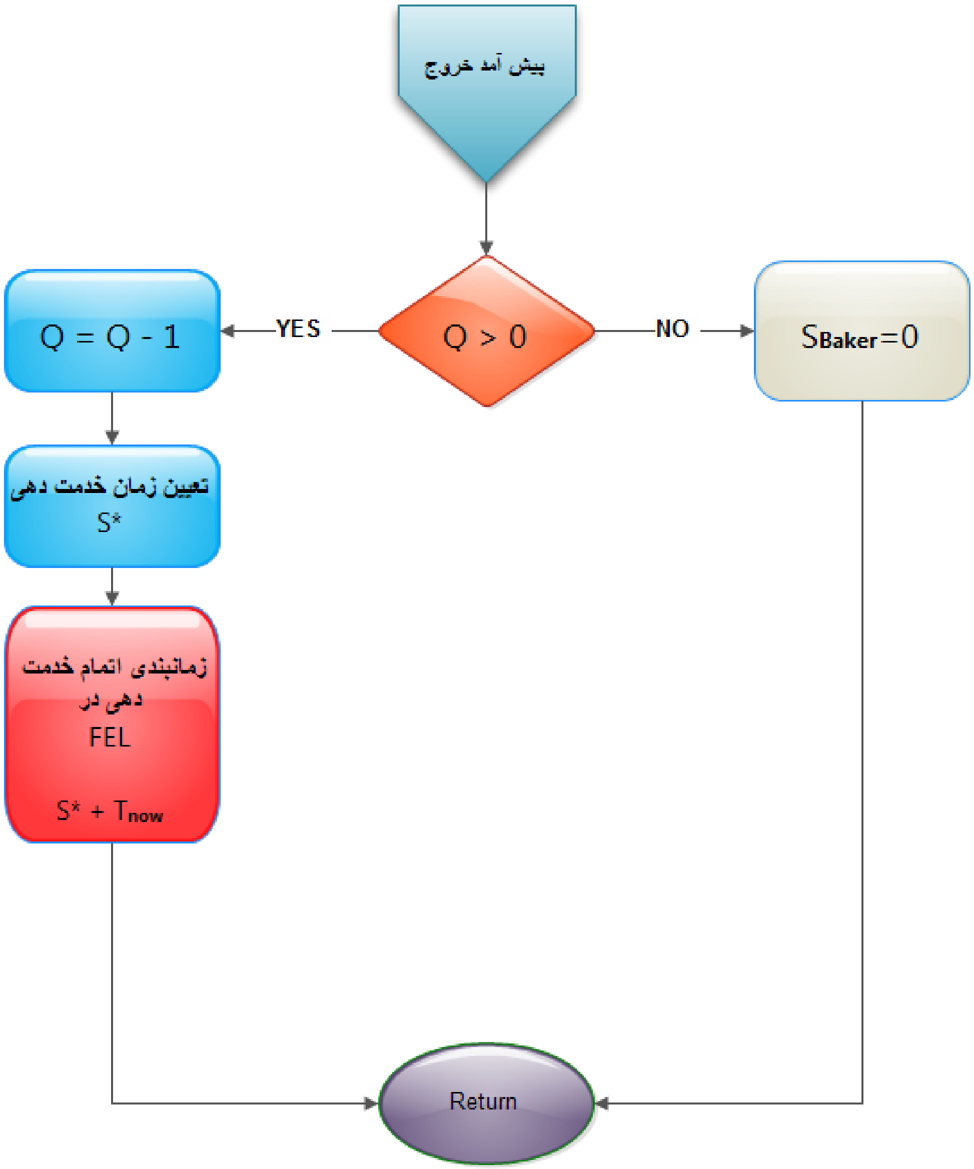
Simulation Controller Flow Chart



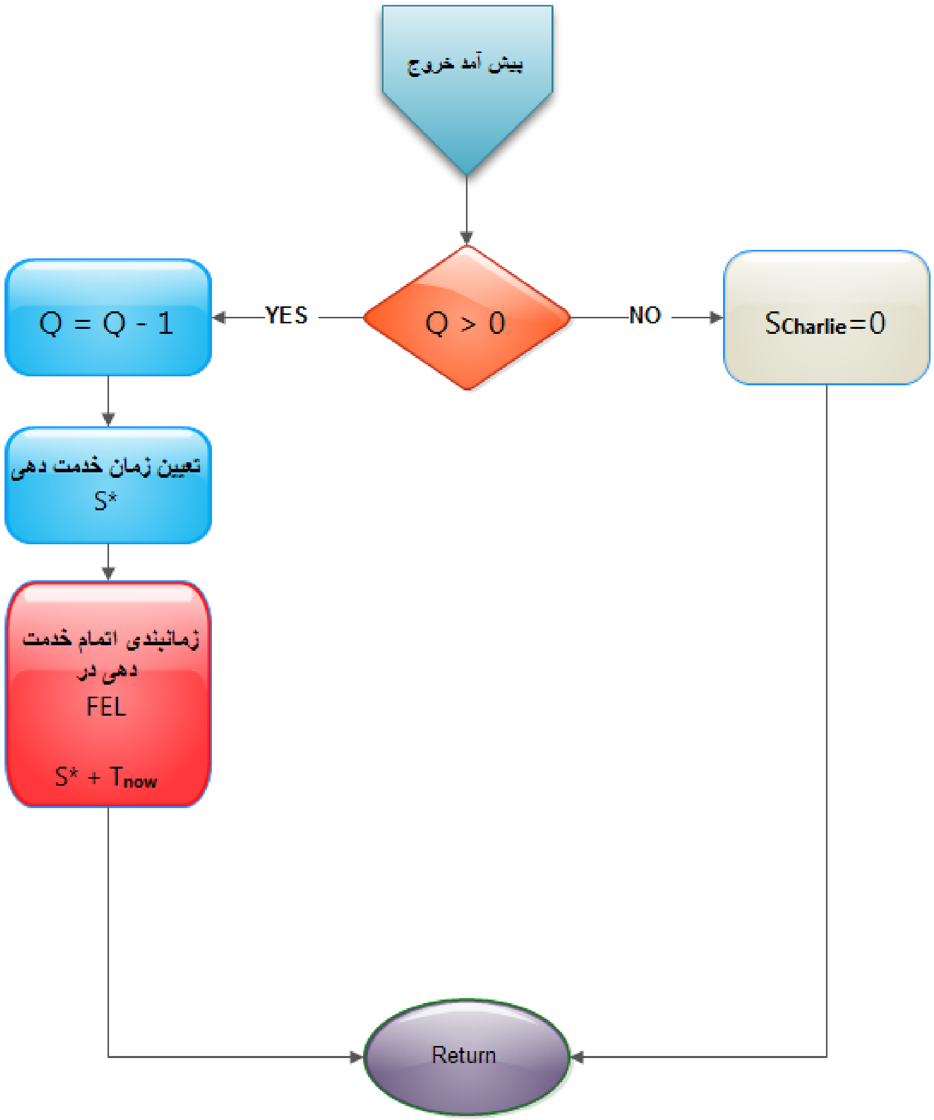
Simulation Customers Arrival Event - Flow Chart



Simulation Able's Customers Departure Flow Chart



Simulation Baker's Customers Departure Flow Chart



Simulation Charlie's Customers Departure Flow Chart

randn

Normally distributed pseudorandom numbers

اعداد تصادفی(توزیع نرمال اعداد شبه تصادفی) در #C

همانطور که می دانید؛ در کتابخانه های زبان #C، تابع randn وجود ندارد. برای رفع این کاستی روش های گوناگونی وجود داشت، استفاده از کتابخانه های بسیار قدرتمند و گسترده در زمینه محاسبات عددی که به زبان #C پیاده سازی شده، و یا استفاده از توابع نرم افزار Matlab در زبان #C. ولی روشی که ما برگزیدیم؛ پیاده سازی یک الگوریتم برای تولید اعداد تصادفی با خاصیت مورد نظر بود.

روش Box – Muller[۱]

در حال حاضر، برنامه برای تولید اعداد شبه تصادفی از این الگوریتم استفاده می نماید. اعداد تولیدی با میانگین ۰ و واریانس ۱ هستند.

برای بدست آوردن مقادیر در بازه های دلخواه، از الگوی زیر پیروی می کنیم:

اعداد تصادفی با میانگین ۱۰ و انحراف معیار ۵ تولید کنید

```
var Mean = 10;
var Deviation = 5;
return Mean + (Deviation * GetNormal());
```

در پروژه های بعدی تیم علاقه بسیاری دارد از الگوریتم Ziggurat برای تولید اعداد تصادفی استفاده نمایید. این الگوریتم بسیار سریع تر از الگوریتم فعلی می باشد.

کد مربوط به تولید اعداد تصادفی (نرمال) توسط John D. Cook پیاده سازی شده و به صورت متن باز در اختیار عموم قرار داده شده است. به دلیل احترام به جامعه متن باز اسم ایشان در برنامه ذکر شده است.

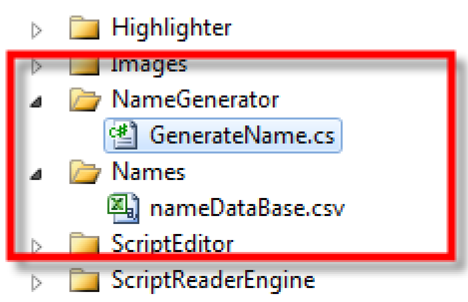
[1] در ضمیمه این گزارش، روش کار این الگوریتم به صورت کامل شرح داده شده است.

```
// Get normal (Gaussian) random sample with mean 0 and standard deviation 1
public static double GetNormal()
{
    // Use Box-Muller algorithm
    double u1 = GetUniform();
    double u2 = GetUniform();
    double r = Math.Sqrt(-2.0*Math.Log(u1));
    double theta = 2.0*Math.PI*u2;
    return r*Math.Sin(theta);
}

// Get normal (Gaussian) random sample with specified mean and standard deviation
public static double GetNormal(double mean, double standardDeviation)
{
    if (standardDeviation <= 0.0)
    {
        string msg = string.Format("Shape must be positive. Received {0}.", standardDeviation);
        throw new ArgumentOutOfRangeException(msg);
    }
    return mean + standardDeviation*GetNormal();
}
```

اسامی تصادفی!

در طراحی شبیه سازی مسئله دوم، گروه دست به ابتکاری جالب زده است و برای هر یک از مشتریان، یک نام به صورت تصادفی تولید می کند. این کار صرفا جهت بالا بردن کارایی نرم افزار شبیه ساز است.

برای تولید اسامی تصادفی، نرم افزار از یک بانک اطلاعاتی در فورمت CSV استفاده می کند. این بانک شامل ۱۰,۰۰۰ نام و نام خانوادگی می باشد.



No.	First Name	Last Name	Arrival Time	Service Time	Queue Time	Departure Time	Service Provider
1	Carl	Horne	0	7	0	7	Able
2	Herbert	Abbott	3	10	0	13	Baker
3	Neal	Lewis	5	11	0	16	Charlie
4	Vincent	Phelps	6	16	1	23	Able
5	Zachary	Eason	10	4	3	17	Baker
6	Edwin	Cline	10	5	6	21	Charlie

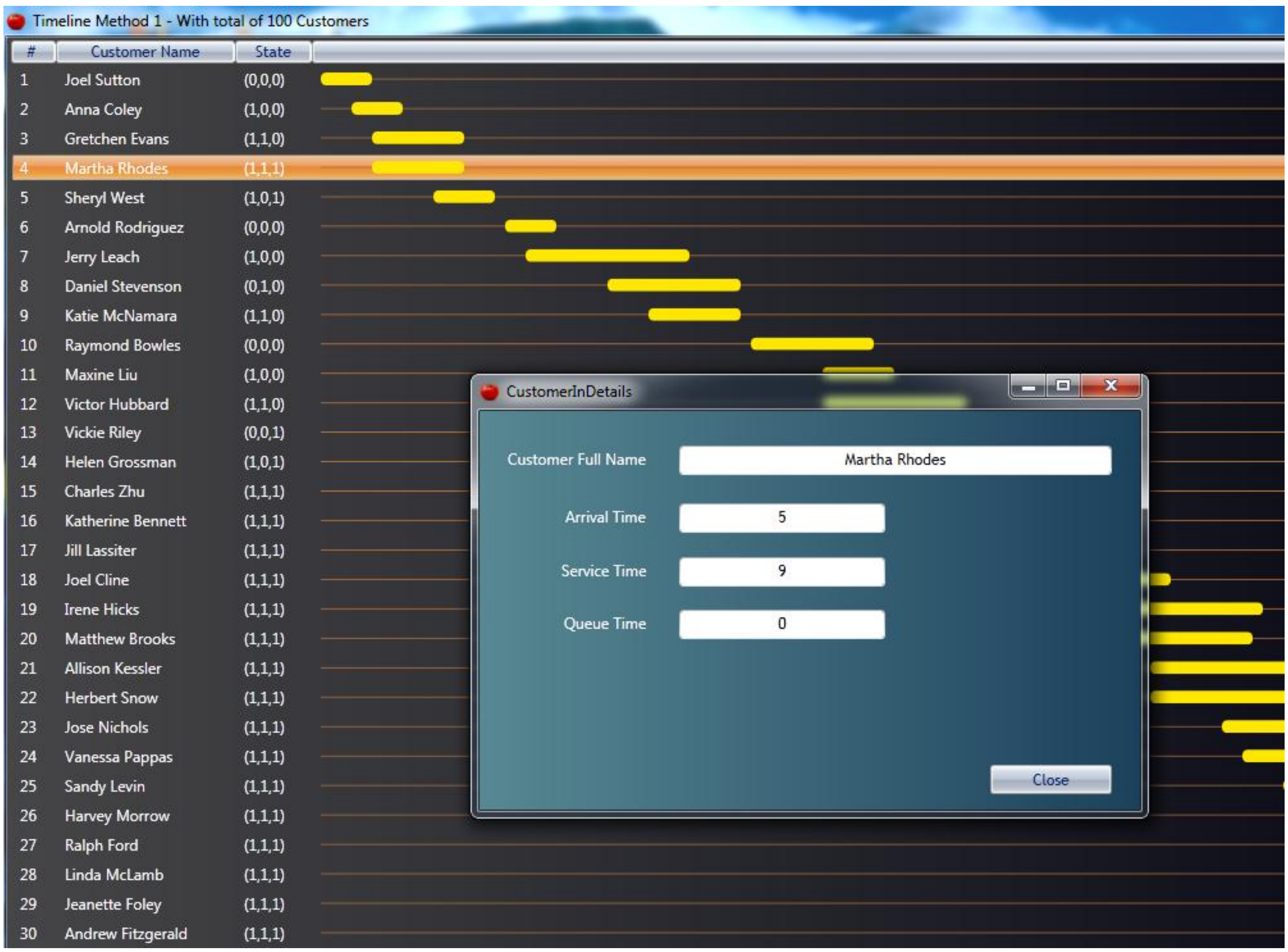
در صفحه ی بعد، کلاس مربوط به تولید اسامی تصادفی را می توانید مشاهده کنید.

```csharp
public class GenerateName
{
    private readonly List<string> _firstNames;
    private readonly List<string> _lastNames;

    private readonly Random _randomEngine;

    public GenerateName()
    {
        _randomEngine = new Random(DateTime.Now.Millisecond);

        _firstNames = new List<string>();
        _lastNames = new List<string>();
        try
        {
            var reader =
                new StreamReader(new FileStream(@"Names\nameDataBase.csv",
                    FileMode.Open,FileAccess.Read));

            string fileSt = reader.ReadToEnd();
            fileSt = fileSt.Replace("\r\n", ";");
            String[] values = fileSt.Split(';');
            foreach (string val in values)
            {
                string[] str = val.Split(',');
                if (str.Length == 2)
                {
                    _firstNames.Add(str[0]);
                    _lastNames.Add(str[1]);
                }
            }
        }
        catch (Exception)
        {
            throw new Exception("Could not Initialize the Calss!");
        }
    }

    public string GenerateFirstName()
    {
        return _firstNames[GenerateNumber()];
    }

    public string GenerateLastName()
    {
        return _lastNames[GenerateNumber()];
    }

    public int GenerateNumber()
    {
        int minimum = _lastNames.Count;
        if (_firstNames.Count < minimum)
            minimum = _firstNames.Count;

        int returnValue = _randomEngine.Next(0, minimum);
        return returnValue;
    }
}
```

Timeline!

تایم لاین!

در مراحل پایانی پیاده سازی پروژه؛ ایده ای به ذهن گروه خطور کرد! نشان دادن زمان شبیه سازی به صورت گرافیکی!



ضمیمه ۱

الگوریتم Box-Muller

Box Muller Method

Takashi Shinzato(shinzato@sp.dis.titech.ac.jp)

January 27, 2007

1 Motivation

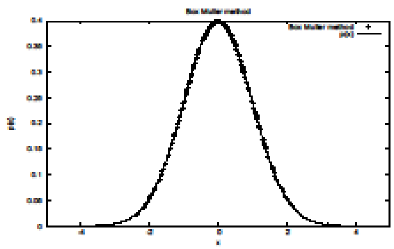


Figure 1: Box Muller method; random variable x is asymptotically distributed on $N(0,1)$.

In this informal note, we discuss normal gaussian random variable $x \sim N(0,1)$[1] using stochastic variables on uniform distribution in $[0,1]$. At first step, we indicate then the following relation as

$$\int_{-\infty}^{\infty} dx e^{-\frac{x^2}{2}} = \sqrt{2\pi}. \tag{2}$$

In order to prove eq.(2), we present via

$$\left(\int_{-\infty}^{\infty} dx e^{-\frac{x^2}{2}}\right)^2 = \int_{-\infty}^{\infty} dx dy e^{-\frac{x^2+y^2}{2}}, \tag{3}$$

and replace x and y to r and θ like[2]

$$x = r\cos\theta, \tag{4}$$
$$y = r\sin\theta. \tag{5}$$

Using eq.(4) and eq.(5), they are also held as

$$x^2+y^2 = r^2 \tag{6}$$
$$\tan^{-1}\frac{y}{x} = \theta \tag{7}$$
$$\left|\frac{\partial(x,y)}{\partial(r,\theta)}\right| = r \tag{8}$$

[1]Normal distribution is distributed on

$$p(x) := \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}} \Longleftarrow N(0,1) \tag{1}$$

with the mean 0 and the variance 1.

[2]$r > 0$ and $0 \le \theta \le 2\pi$

Eq.(8) is Jaccobian. Hence, we show so

$$\begin{aligned}\int_{-\infty}^{\infty} dx dy e^{-\frac{x^2+y^2}{2}} &= \int_0^{2\pi} d\theta \int_0^{\infty} dr r e^{-\frac{r^2}{2}} \\ &= 2\pi \int_0^{\infty} du e^{-u} \quad \left(u = \frac{r^2}{2}\right) \\ &= 2\pi \left[-e^{-u}\right]_0^{\infty} = 2\pi. \end{aligned}\tag{9}$$

Namely, eq.(2) is true. Therefore, we give then

$$1 = \frac{1}{2\pi}\int_{-\infty}^{\infty} dx dy e^{-\frac{x^2+y^2}{2}}, \tag{10}$$

and in the case that x is distributed on probability measure (or mass probability) $p(x) = \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}}$, stochastic variable x is called gaussian random variable.

2 Box Muller Method

In this section, we present and introduce Box Muller method. Providing with the law of large numbers or center limit theorem, gaussian random variable has played an important role and one makes usefull for hypothesis and test in Data science and so on. We define a new function as

$$U(R) := \frac{1}{2\pi}\int_{x^2+y^2\le R^2} dx dy e^{-\frac{x^2+y^2}{2}}. \tag{11}$$

This integral interval is in $x^2+y^2 \le R^2$. We calcurate then briefly

$$\begin{aligned}U(R) &= \frac{1}{2\pi}\int_0^{2\pi} d\theta \int_0^R dr r e^{-\frac{r^2}{2}} \\ &= \int_0^{\frac{R^2}{2}} du e^{-u} = 1 - e^{-\frac{R^2}{2}} \end{aligned}\tag{12}$$

$U(R)$ is nondecreasing function satisfied as

$$\lim_{R\to 0} U(R) = 0 \tag{13}$$
$$\lim_{R\to\infty} U(R) = 1 \tag{14}$$

Moreover, by eq.(11), the integral interval is replaced using $0 \le r \le R$ and eq.(4) and eq.(5). We represent that using $p \in [0,1]$ and eq.(12), variable R of $U(R) = p$ is decided as

$$U(R) = p \Longrightarrow R = \sqrt{-2\log(1-p)}. \tag{15}$$

In setting $s := 1-p \in [0,1]$ and $t \in [0,1]$, it is given as

1

boxmuller.pdf

$$x = \begin{cases} \sqrt{-2\log(s)}\cos(2\pi t) \\ \sqrt{-2\log(s)}\sin(2\pi t)\,. \end{cases} \tag{16}$$

That is to say, we give then the scheme that we generate gaussian random variable distributed on $N(0,1)$ using uniform distribution on $[0,1]$; its scheme is called Box Muller method. More being generalized, stochastic variable $z \sim N(\mu, \sigma^2)$, with the mean μ and the variance σ^2, is given as

$$z = \mu + \sigma\sqrt{-2\log(s)}\cos(2\pi t)\,, \tag{17}$$

or

$$z = \mu + \sigma\sqrt{-2\log(s)}\sin(2\pi t)\,. \tag{18}$$

3 Conclusion

We introduce and discuss, in this informal note, the scheme of normal distribution $N(0,1)$ using unique distribution on $[0,1]$, Box Muller method.

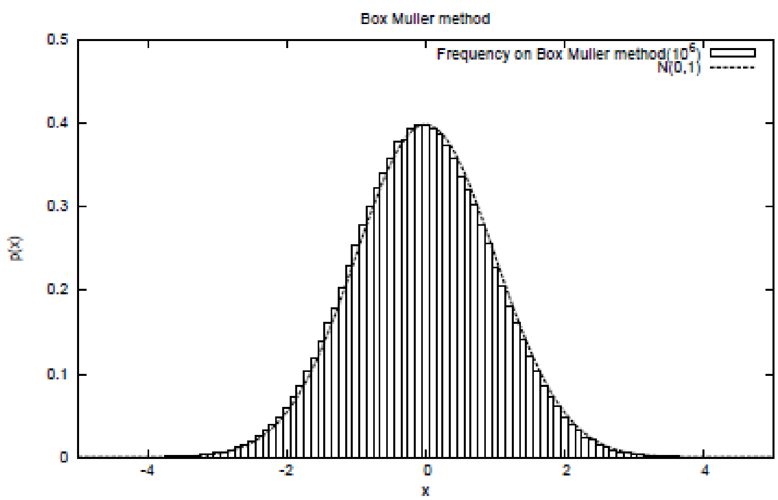


Figure 2: Frequency on Box Muller method is asymptotically distributed on $N(0,1)$.

boxmuller.pdf