



دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)

دانشکده مهندسی صنایع و سیستم‌های مدیریت

## تمرین یکم درس اصول شبیه‌سازی

اعضا:

حامد اعراب - ۹۹۲۵۰۰۳

شهریار خلوتی - ۹۹۲۵۰۱۵

استاد: دکتر عباس احمدی

تدریس‌یار: مهدی محمدی

بهار ۱۴۰۳

## فهرست

مسائل	۱
مسئله یکم	۱
مسئله دوم	۱
مقدمه	۲
چارچوبی برای شبیه‌سازی رویدادهای گسسته	۳
کنترلر	۴
رویداد	۵
پیاده‌سازی و حل مسئله‌ها	۷
حالت کلی: چند خدمت‌دهنده	۷
موجودیت‌های مختص به این سیستم	۷
کنترلر	۷
رویدادها	۸
حالت خاص: دو خدمت‌دهنده	۱۰
کنترلر	۱۰
نتایج	۱۱
مسئله یکم	۱۱
مسئله دوم	۱۱
نتیجه‌گیری	۱۲
جدول گزارش عملکرد اعضای گروه	۱۳



## مسائل

### مسئله یکم

آریا و بردیا دو خدمت‌دهنده در یک رستوران خودرویی هستند. فاصله میان ورود هر دو خودرو یکنواخت و در بازه ۳ تا ۶ دقیقه است. زمان خدمت آریا و بردیا به هر خودرو، یکنواخت و در بازه ۹ تا ۱۲ دقیقه است.

- (۱) نمودارهای جریان را برای کنترلر و رویدادهای شبیه‌سازی بکشید.
- (۲) سیستم را برای ۹ ساعت شبیه‌سازی کنید. در حالت اول، هر بار نخستین خدمت‌دهنده در دسترس را برای خدمت‌دهی، و در حالت دوم، یک خدمت‌دهنده در دسترس را تصادفی انتخاب کنید.
- (۳) میزان به‌کارگیری (درصد زمان اشتغال) آریا و بردیا را در هر حالت تخمین بزنید.
- (۴) میانگین زمان انتظار هر مشتری را در هر حالت به‌دست آورید.
- (۵) در هر حالت چند نفر وارد صف انتظار می‌شوند؟

### مسئله دوم

در یک سیستم خدماتی با چندین ایستگاه موازی (I خدمت‌دهنده)، زمان میان ورود هر دو مشتری یکنواخت و در بازه ۰ تا ۴ دقیقه، و زمان خدمت هر خدمت‌دهنده به هر مشتری، یکنواخت و در بازه ۶ تا ۱۰ دقیقه است.

- (۱) نمودارهای جریان را برای کنترلر و رویدادهای شبیه‌سازی بکشید.
- (۲) سیستم را برای ۸ ساعت شبیه‌سازی کنید.
- (۳) تعداد بهینه خدمت‌دهنده‌ها را به‌دست آورید، به طوری که میانگین زمان انتظار کمینه و درصد زمان اشتغال هر خدمت‌دهنده بیشینه شود.

## مقدمه

مسئله‌هایی که پیش‌تر آمده‌اند، در بنیان خود، شبیه‌سازی رویدادهای گسسته<sup>۱</sup> را بیان می‌کنند. در مسئله نخست، سیستمی داریم با یک صف<sup>۲</sup> و دو خدمت‌دهنده<sup>۳</sup> که در کنار هم به مشتریان<sup>۴</sup> خدمت می‌دهند. رویدادهای<sup>۵</sup> سیستم نیز به ورود<sup>۶</sup> و خروج<sup>۷</sup> مشتریان محدود می‌شوند. در مسئله دوم، حالت کلی مسئله پیشین را داریم، یک صف، چندین خدمت‌دهنده موازی<sup>۸</sup>، و همان رویدادها. حال، باید به این نکته نیز بپردازیم که علی‌رغم شباهت سیستم‌های این دو مسئله، خواسته‌های هر مسئله متفاوت است. در مسئله نخست، باید سیستم را با دو روش متفاوت در انتخاب خدمت‌دهنده در دسترس<sup>۹</sup> شبیه‌سازی کنیم. سپس با به‌دست آوردن و مقایسه چند معیار کارایی<sup>۱۰</sup>، مانند درصد زمان اشتغال خدمت‌دهنده‌ها، روش برتر را بیابیم. اما در مسئله دوم، صرفاً باید تعداد بهینه‌ی خدمت‌دهنده‌ها را با توجه به مقادیر داده‌شده برای زمان‌های میان هر رویداد بیابیم، به طوری که میانگین زمان انتظار کمینه<sup>۱۱</sup> و درصد زمان اشتغال هر خدمت‌دهنده بیشینه<sup>۱۲</sup> شود.

---

<sup>1</sup> Discrete-Event Simulation

<sup>2</sup> Queue

<sup>3</sup> Server

<sup>4</sup> Customers

<sup>5</sup> Events

<sup>6</sup> Arrival

<sup>7</sup> Departure

<sup>8</sup> Parallel

<sup>9</sup> Available

<sup>10</sup> Performance Metrics

<sup>11</sup> Minimized

<sup>12</sup> Maximized

## چارچوبی برای شبیه‌سازی رویدادهای گسسته

حال، با توجه به این که (۱) هر دو مسئله، شبیه‌سازی رویدادهای گسسته را در خود گنجانده‌اند و (۲) احتمالاً در تمرین‌های آینده نیز با این نوع مسائل مواجه خواهیم شد، تصمیم گرفتیم که یک چارچوب<sup>۱۳</sup> و بنیان واحد برای حل آن‌ها طراحی و پیاده کنیم. این چارچوب باید ویژگی‌های زیر را داشته باشد:

- (۱) گسترش‌پذیری<sup>۱۴</sup>: در مسائل گوناگون با سیستم‌هایی سروکار داریم که متغیرها<sup>۱۵</sup>، وضعیت<sup>۱۶</sup>، منطق<sup>۱۷</sup>، و رفتار<sup>۱۸</sup> مخصوص خود را دارند. بنابراین، به چارچوبی نیاز داریم که حداقل امکانات شبیه‌سازی را فراهم کرده و اجازه‌ی افزودن سایر اجزای مورد نیاز را بدهد.
  - (۲) مقیاس‌پذیری<sup>۱۹</sup>: مسائل شبیه‌سازی در پیچیدگی<sup>۲۰</sup> و اندازه با هم تفاوت دارند. بنابراین، چارچوب مورد استفاده باید قابلیت به‌کارگیری را در مقیاس‌های مختلف داشته باشد.
  - (۳) بهره‌وری<sup>۲۱</sup>: چارچوب ما باید در اجرای فرآیند شبیه‌سازی سریع بوده و مصرف منابع کمی داشته باشد. زیرا با داشتن بهره‌وری مناسب می‌توان یک سیستم را به دفعات بیش‌تری شبیه‌سازی کنیم و از صحت خروجی‌ها اطمینان حاصل کنیم.
- با توجه به این ویژگی‌ها، چارچوب ما دو موجودیت کنترلر<sup>۲۲</sup> و رویداد را برای استفاده فراهم می‌کند. در ادامه، به جزئیات آن‌ها بیش‌تر می‌پردازیم.

---

<sup>13</sup> Framework  
<sup>14</sup> Extensibility  
<sup>15</sup> Variables  
<sup>16</sup> State  
<sup>17</sup> Logic  
<sup>18</sup> Behavior  
<sup>19</sup> Scalability  
<sup>20</sup> Complexity  
<sup>21</sup> Efficiency  
<sup>22</sup> Controller

## کنترلر

کنترلر، وظیفه‌ی اجرا و مدیریت فرآیند شبیه‌سازی را بر عهده دارد. به زبان ساده، با شروع فرآیند شبیه‌سازی، رویدادها به کنترلر ارسال می‌شوند، و کنترلر آن‌ها را به «فهرست رویدادهای آینده<sup>۲۳</sup>» می‌افزاید و این فهرست را بر اساس «زمان مقرر<sup>۲۴</sup>» رویدادها مرتب می‌کند. پس از پایان شلیک<sup>۲۵</sup> رویداد پیشین، رویداد پیشرو<sup>۲۶</sup> از FEL خارج و پس از جلو بردن ساعت شبیه‌سازی<sup>۲۷</sup>، شلیک می‌گردد. در این میان اگر ساعت شبیه‌سازی به زمان مشخص شده برای پایان برسد یا از آن فراتر رود، شبیه‌سازی متوقف می‌گردد. کنترلر، دو پارامتر رویداد نخستین<sup>۲۸</sup> و زمان پایان شبیه‌سازی<sup>۲۹</sup> را دریافت و صحت‌سنجی<sup>۳۰</sup> می‌کند. سنجش صحت پارامترها عبارتند از مثبت بودن زمان پایان شبیه‌سازی و صفر بودن زمان مقرر رویداد نخستین. پس از سنجش صحت پارامترها، رویداد نخستین به FEL افزوده شده و ساعت شبیه‌سازی برابر با دقیقه صفر قرار داده می‌شود. لازم به ذکر است که در این چارچوب، تمامی زمان‌ها با واحد دقیقه به کار می‌روند.

حال، کنترلر برای شروع فرآیند شبیه‌سازی آماده است. با شروع فرآیند، رویداد نخستین که به FEL افزوده شده بود را از این فهرست خارج و سپس شلیک می‌کنیم. در پی آن، رویدادهای بعدی دریافت شده و به FEL افزوده می‌گردند و این حلقه تکرار می‌گردد، تا این که به زمان پایان شبیه‌سازی برسیم. در شکل بعدی، نمودار جریان را برای فرآیند شبیه‌سازی کنترلر می‌توانید مشاهده کنید. این نمودار همان توضیحات ساده‌ای که پیش‌تر داده شدند را با جزئیات بیش‌تر ارائه می‌دهد.

---

<sup>23</sup> Future Events List (FEL)

<sup>24</sup> Due Time

<sup>25</sup> Triggering

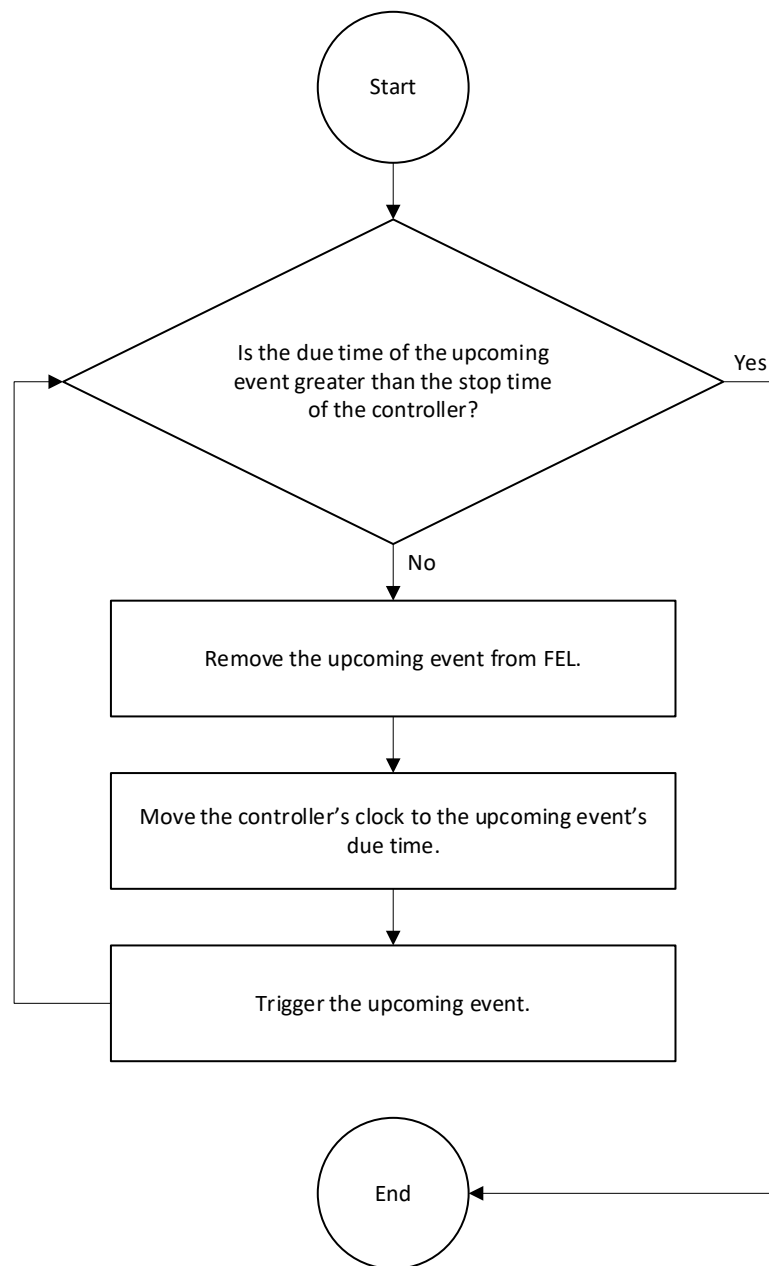
<sup>26</sup> Upcoming Event

<sup>27</sup> Simulation Clock

<sup>28</sup> Initial Event

<sup>29</sup> Simulation Stop Time

<sup>30</sup> Validation



## رویداد

این موجودیت، بستر را برای ایجاد رویدادهای مختلف یک سیستم فراهم می‌کند. تنها پارامتری که دریافت می‌کند فاصله<sup>۳۱</sup> زمان شلیک آن با ساعت شبیه‌سازی است و پس از آن که به کنترلر ارسال شد، زمان مقرر مشخص می‌گردد. در نهایت، کنترلر رویداد را به FEL افزوده و فهرست بر اساس زمان مقرر مرتب می‌کند.

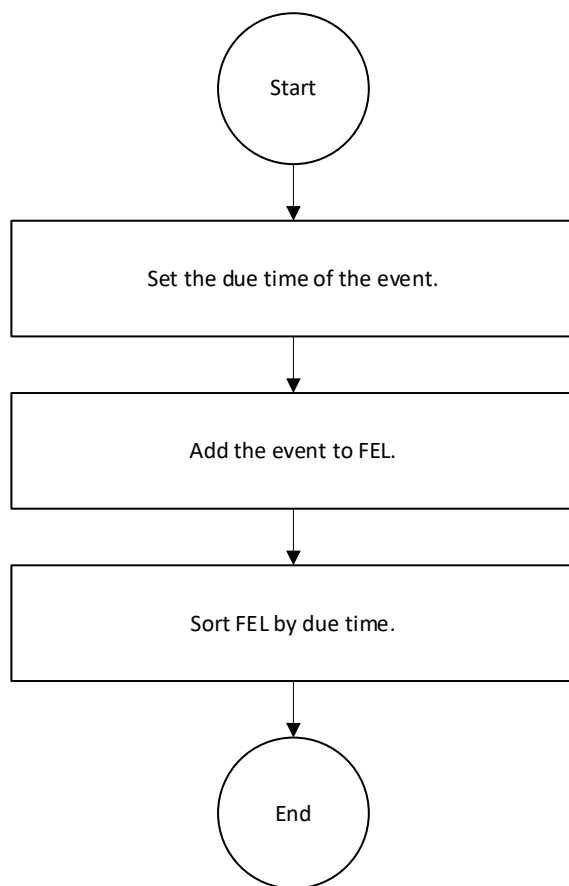
<sup>31</sup> Interval



چارچوبی برای شبیه‌سازی رویدادهای گسسته

$$Due\ Time_{event} = Simulation\ Clock + Interval_{event}$$

در ادامه، نمودار جریان را برای ارسال یک رویداد به کنترلر شبیه‌سازی مشاهده می‌کنید.



در ادامه، به پیاده‌سازی جواب هر مسئله با استفاده از این چارچوب می‌پردازیم.

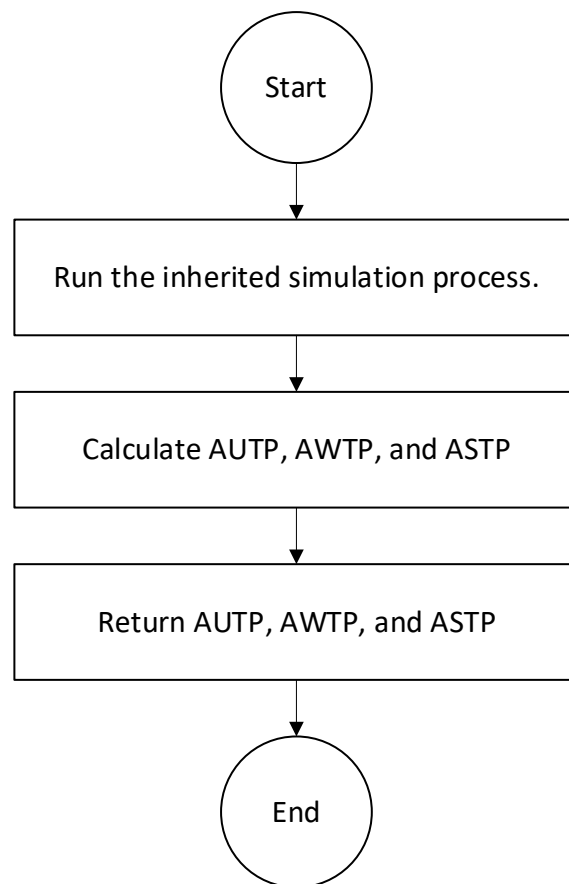
## پیاده‌سازی و حل مسئله‌ها

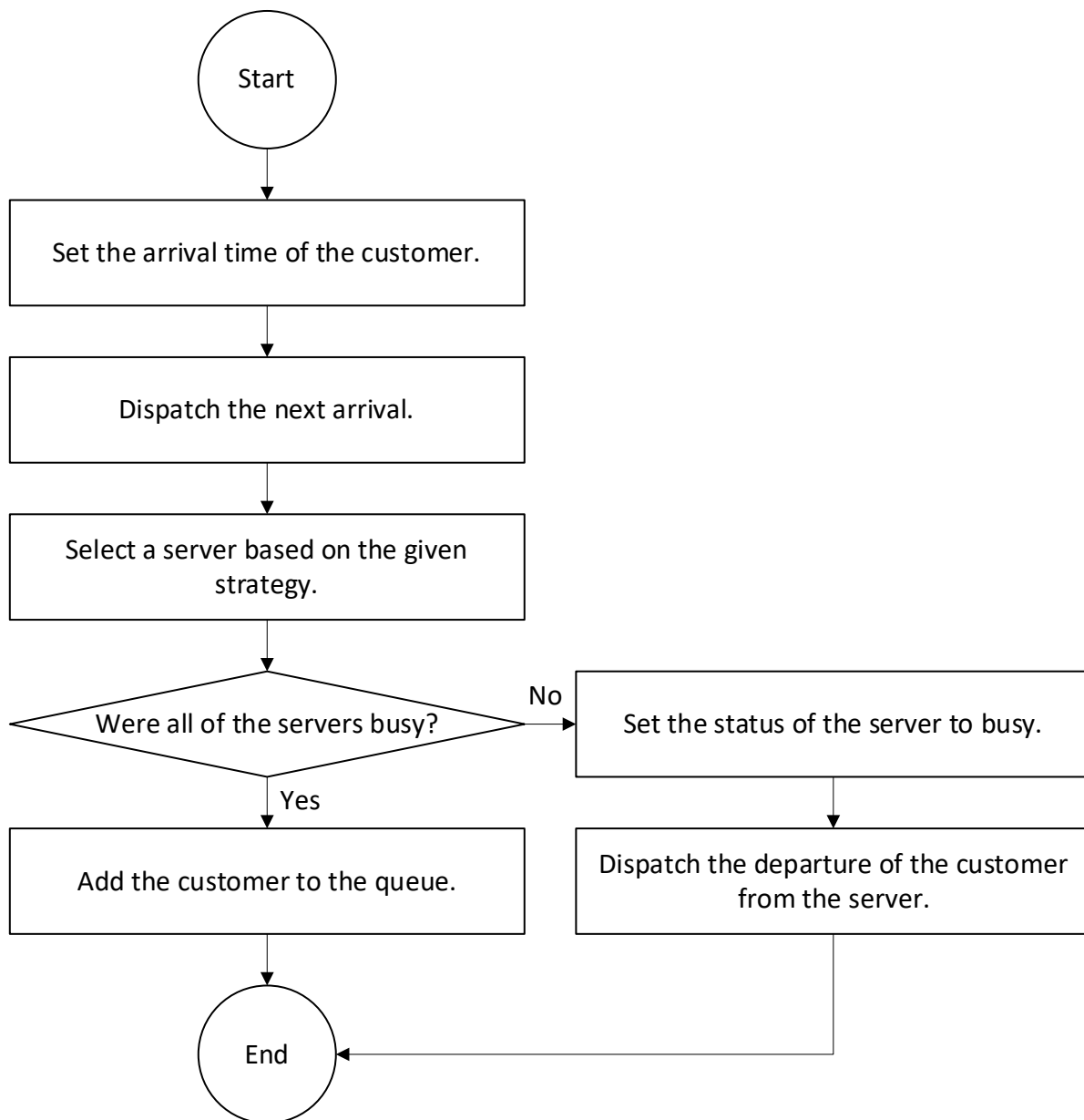
از آنجایی که سیستم نخست حالتی خاص از سیستم دوم است، ابتدا حالت کلی را پیاده کرده و برای پیاده‌سازی حالت خاص از مفهوم «ارث‌بری» استفاده می‌کنیم.

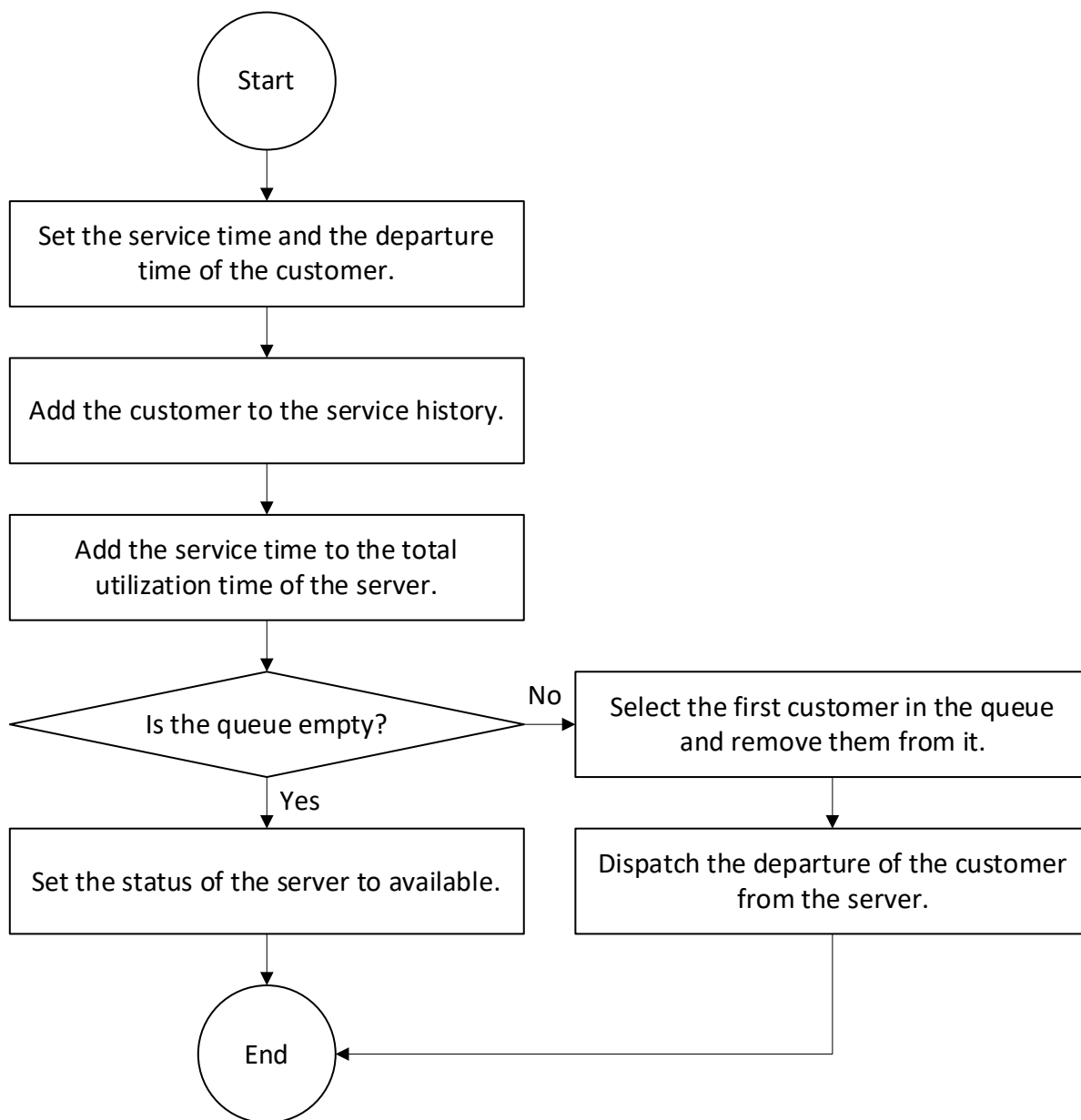
### حالت کلی: چند خدمت‌دهنده

موجودیت‌های مختص به این سیستم

کنترلر

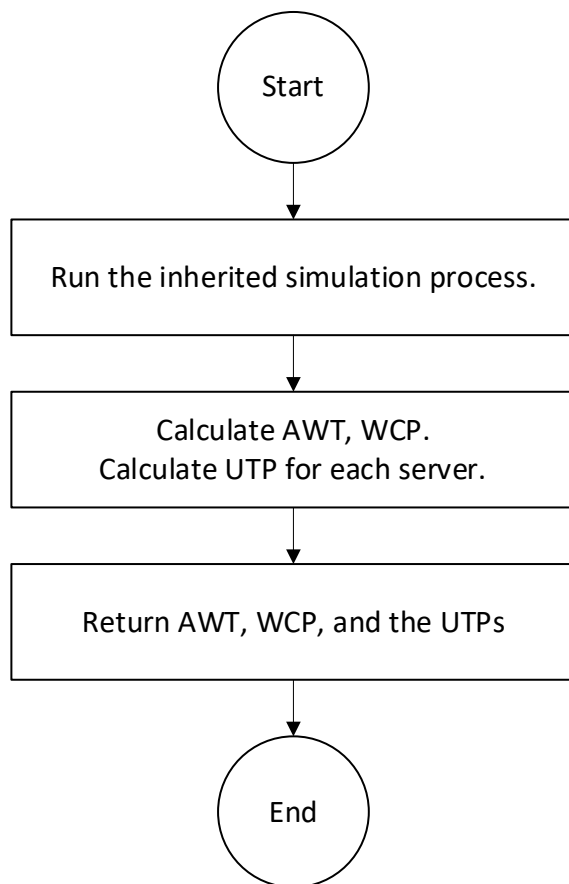






## حالت خاص: دو خدمت‌دهنده

کنترلر



## نتایج

مسئله یکم

مسئله دوم

نتیجه گیری

نتیجه گیری

جدول گزارش عملکرد اعضای گروه

جدول گزارش عملکرد اعضای گروه