

باسمه تعالی
دانشگاه علم و صنعت ایران
دانشکده مهندسی کامپیوتر
درس سیستم‌های توزیع شده (پاییز ۹۲)
تمرین شماره ۱
مهلت انجام: ۹۲/۸/۲۴

مونیتورینگ سیستم‌های توزیع شده

در این تمرین قرار است مونیتورینگ یک سیستم توزیع شده پیاده سازی شود. برای مدیریت سیستم‌های توزیع شده نیاز است اطلاعات آماری مجموع سیستم‌ها جمع آوری شود.

مونیتورینگ در سیستم بوسیله‌ی یک عامل¹ انجام می‌شود. برای پیاده‌سازی یک عامل ابتدا لازم است متریک‌های مختلف سیستم جمع آوری شود. متریک‌هایی که برای تصمیم گیری انتخاب شده‌اند با واحد مورد نظر به شرح زیر است (به این قسمت metric gatherer یا به اختصار MG گفته می‌شود):

- CPU Usage - درصد
- Memory Usage - درصد
- File System Write - بلوک بر ثانیه
- File System Read - بلوک بر ثانیه

در این عامل‌ها، چهار ریسمان ایجاد می‌شود که هر کدام وظیفه‌ی جمع‌آوری داده یکی از متریک‌های فوق را به عهده دارند. سپس این متریک‌های از طریق سوکت به یک سرور خارجی ارسال می‌شوند. این سرور خارجی نیز یک پردازش است که اطلاعات ارسالی از سوی یک یا چند عامل را تجمیع می‌کند و در قالب یک سری اطلاعات به سرور اصلی می‌فرستد. برای ایجاد پردازش‌های عامل، دو ورودی IP سرور و یک نام یکتا برای عامل داده می‌شود. هر یک از ریسمان‌های در حال اجرا در عامل، اطلاعات متریک مربوط به خود را به همراه نام عامل به سرور مربوطه می‌فرستد. سرور مذکور به ازای هر کلاینت یک ریسمان برای جمع‌آوری اطلاعات ایجاد می‌کند و اطلاعات رسیده را برای پردازش به ریسمان مربوطه می‌سپارد.

خروجی برنامه سرور به ازای یکی از عامل‌ها می‌تواند به این صورت باشد که هر پنج ثانیه هرچهار MG را نشان دهد:

1 agent

باسمه تعالی
دانشگاه علم و صنعت ایران
دانشکده مهندسی کامپیوتر
درس سیستم‌های توزیع شده (پاییز ۹۲)
تمرین شماره ۱
مهلت انجام: ۹۲/۸/۲۴

```
at 17:25:23
CPU usage: 23%
Memory usage: 76%
FS write: 47623 B/sec
FS Read: 5342 B/sec
=====
at 17:25:28
CPU usage: 26%
Memory usage: 75%
FS write: 44183 B/sec
FS Read: 6410 B/sec
=====
at 17:25:33
CPU usage: 24%
Memory usage: 76%
FS write: 46298 B/sec
FS Read: 9823 B/sec
```

در مورد نشان دادن یک مقطع زمانی امکان اینکه هر چهار پیام با یک زمان نشانه‌گذاری شده باشند خیلی کم است. به همین خاطر در هنگام نشان دادن متریک‌ها، آخرین پیغام رسیده باید به عنوان مقدار در آن زمان خاص لحاظ شود.

در نهایت یک پرده واحد سرور وجود دارد که تمامی اطلاعات را از سرورهای میانی دریافت می‌کند و برای همه‌ی عامل‌ها، اطلاعات به روز شده را نگهداری می‌نماید. این سرور نیز دو فقره اطلاعات در مورد هر متریک هر عامل نگهداری می‌کند: نخست آخرین مقدار مشاهده شده‌ی آن متریک و دوم میانگین مقادیر مشاهده شده برای آن متریک از آغاز به کار سیستم.

در این سرور می‌بایست یک کنسول پیاده‌سازی گردد که دستورات زیر را اجرا نماید:

- Show : نام تمامی عامل‌های فعال را چاپ می‌نماید.
- Kill \$agent : دستور پایان کار عامل را برای او از طریق سرورهای میانی ارسال می‌نماید.
- Now \$agent : چاپ آخرین وضعیت عامل.
- Avg \$agent : چاپ میانگین متریک‌های عامل.
- NowAll : چاپ آخرین وضعیت تمامی عامل‌ها.

باسمه تعالی
دانشگاه علم و صنعت ایران
دانشکده مهندسی کامپیوتر
درس سیستم‌های توزیع شده (پاییز ۹۲)
تمرین شماره ۱
مهلت انجام: ۹۲/۸/۲۴

• AvgAll : چاپ میانگین وضعیت متریک‌های تمامی عامل‌ها.

یک نمونه از اجرای این دستورات در سرور مشابه زیر می‌باشد:

```
Show
=====
client1
client2
=====
Now client1
=====
at 17:25:23
CPU usage: 23%
Memory usage: 76%
FS write: 47623 B/sec
FS Read: 5342 B/sec
=====
Avg client1
=====
at 17:24:43 to 17:25:23
CPU usage: 22%
Memory usage: 63%
FS write: 6421 B/sec
FS Read: 1642 B/sec
=====
NowAll
=====
at 17:25:23
CPU usage: 26%
Memory usage: 51%
FS write: 4943 B/sec
FS Read: 1723 B/sec
=====
AvgAll
=====
at 17:22:12 to 17:25:23
CPU usage: 8%
Memory usage: 54%
FS write: 4523 B/sec
FS Read: 1295 B/sec
=====
kill client1
```

باسمه تعالی
دانشگاه علم و صنعت ایران
دانشکده مهندسی کامپیوتر
درس سیستم‌های توزیع شده (پاییز ۹۲)
تمرین شماره ۱
مهلت انجام: ۹۲/۸/۲۴

نکات پیاده سازی:

- ریسمان‌ها به وسیله کتابخانه‌ی PThread پیاده سازی شوند.
- سیگنال‌هایی که امکان رخداد آن‌ها می‌رود باید پیاده سازی و مدیریت شوند و تشخیص سیگنال‌های مورد نیاز برعهده‌ی شماست.
- پروتکل ارتباطی بین عامل و سرورهای تجمیع‌کننده به صورت زیر است:

```
<name></name><mg name>$MG</mg name><mg content>$CONTENT</mg  
content><mg timestamp>$TS</mg timestamp>
```

برای مثال:

```
<name>client1</name><mg name>cpu</mg name><mg content>23</mg  
content><mg timestamp>17:25:23 </mg timestamp>  
  
<name>client1</name><mg name>mem</mg name><mg content>76</mg  
content><mg timestamp>17:25:23 </mg timestamp>  
  
<name>client1</name><mg name>fsw</mg name><mg content>47623</mg  
content><mg timestamp>17:25:23 </mg timestamp>  
  
<name>client1</name><mg name>fsr</mg name><mg content>5342</mg  
content><mg timestamp>17:25:23 </mg timestamp>
```

برای مثالی که در بالا ذکر شد می‌باشد. نکته‌ی قابل توجه اینکه حتما باید از این پروتکل با جزییات ذکر شده استفاده کنید زیرا باید نسخه‌ی سرور و کلاینت نوشته شده توسط شما با نسخه‌ی گروه‌های دیگر قابل ارتباط باشد که مستلزم پیاده سازی واحدی از این پروتکل است.

- پروتکل ارتباطی سرورهای میانی و سرور اصلی به صورت زیر است:

باسمه تعالی
دانشگاه علم و صنعت ایران
دانشکده مهندسی کامپیوتر
درس سیستم‌های توزیع شده (پاییز ۹۲)
تمرین شماره ۱
مهلت انجام: ۹۲/۸/۲۴

```
<name>$AGENT</name><cpu content>$CONTENT</cpu content><mem  
content>$CONTENT</mem content><fsw content>$CONTENT</fsw content><fsr  
content>$CONTENT</fsr content><mg timestamp>$TS</mg timestamp>
```

به عنوان مثال:

```
<name>client1</name><cpu content>23</cpu content><mem content>76</mem  
content><fsw content>47623</fsw content><fsr content>5342</fsr content><mg  
timestamp>17:25:23 </mg timestamp>
```

- تمام مقادیر متریک‌ها باید درست از سیستم دریافت شود.
- سوالات خود را در فروم درس بپرسید تا دیگران هم استفاده کنند. در ضمن به سوالات مطرح شده توسط دوستان خود -اگر پاسخش را می‌دانید- پاسخ دهید.

نکات تحویل:

- با هر گونه کد مشابه برخورد جدی می‌شود.
- تمام قسمت‌ها باید توسط زبان C نوشته شود و پروژه‌ی شما فقط با gcc کامپایل می‌شود.
- همراه فایل‌های پروژه، می‌بایست گزارشی به زبان فارسی، از نحوه‌ی پیاده‌سازی ارائه گردد.
- گروه‌ها می‌بایست دو نفری باشند.
- تمام فایل‌های مربوط به پروژه را درون یک فولدر با شماره دانشجویی اعضای گروه ریخته و فولدر را zip کنید.