



олимпиада траектория будущего 22-23

Задание полуфинала для номинации Python

Содержание документа:

Формат испытания

Сроки проведения и формат результатов

Описание задания

Теоретическая часть

Требования к способности системы (capabilities)

Требования к качеству кода

- Производительность

- Масштабируемость

- Переносимость

- Доступность

- Надежность

Информационная безопасность

Качество при модернизации

Требования и рекомендации к коду

Критерии оценки

- Соответствие проекта требованиям к способности приложения (capabilities)

- Соответствие проекта требованиям к качеству качества при эксплуатации

- Соответствие проекта требованиям качества при модернизации

Заключительные слова

Рекомендуемые материалы

Приложения

- Пример файла CSV с входными данными

- Пример вывода приложения

Плагат и несамостоятельное выполнение

Определение финалистов

Формат испытания

Формат испытания: Проектная работа. Разработка приложения.

Сроки проведения и формат результатов

Полуфиналисты: Готовые работы необходимо прислать **до 12 марта включительно**. Работы, присланные позже 12 марта не будут рассмотрены.

Участники резерва, приглашенные к участию 13 марта: Готовые работы необходимо прислать **до 20 марта включительно**. Работы, присланные позже 20 марта не будут рассмотрены.

Формат работы: архив с директорией проекта (файлы с исходным кодом модулей и исходные данные в формате csv) или ссылка на репозиторий git.

Результат оформляется как ссылка на облачное хранилище, где вы размещаете составляющие проекта. Ссылку отправляете на ✉ info@tbolimpiada.ru

В теме письма укажите, что ваше письмо для полуфинала по Python.

Описание задания

Вы устроились работать системным администратором в небольшой офис. В ваши обязанности входит мониторинг работоспособности нескольких сайтов. Вы успешно настроили оповещение по почте статуса сервера от хостинга, однако оказалось, что этого недостаточно. И Вы решаете разработать небольшое приложение на Python для того, чтобы оно отправляло Вам сообщения в случае, если какой-либо из сайтов по какой-либо причине станет недоступен.

Теоретическая часть

Доступность сайта обычно определяется клиентом по критерию «открывается сайт в браузере или нет». Однако невозможность открыть страницу сайта может быть обусловлена разными причинами:

1. Отсутствует сетевая связность до сервера:
 - a. IP-адрес сервера недоступен
 - b. DNS сервер не возвращает IP-адрес по доменному имени
 - c. Закрит необходимый порт сервера (443 или 80 для протоколов HTTPS и HTTP соответственно)
2. Не работает сам сервер:
 - a. Служба сервера остановлена
 - b. Сервер неверно сконфигурирован (возвращает код ошибки)
 - c. Сервер перегружен
 - d. Прочие ошибки серверного приложения

3. Проблемы на стороне клиента

Данные о проблемах конкретного клиента (п.3) не могут быть получены и предусмотрены заранее, проблемы серверного приложения (п.2) могут быть отслеживаемыми системой мониторинга хостинга (если эта опция поддерживается и входит в тариф), а проблемы сетевой связности возможно обнаружить с помощью стандартных системных средств ОС или с помощью отдельных модулей Python.

Требования к способности системы (capabilities)

1. Входные данные для приложения должны находится в файле (формат CSV). Пример файла (см. раздел Приложения);
2. В случае, если задано только доменное имя сервера, приложение должно разрешить доменное имя в один или несколько IP-адресов и произвести необходимые проверки для каждого IP-адреса;
3. В случае если разрешить доменное имя сервера не удастся, то приложение должно выводить соответствующее сообщение в вывод;

4. В случае, если задан IP-адрес, приложение производит необходимые проверки для этого адреса;
5. В случае, если входные данные некорректны, приложение при его запуске должно сообщить об этом пользователю и остановить свою работу;
6. В случае, если не указаны порты для проверки, приложение производит только проверку доступности адреса по протоколу ICMP (ping);
7. В случае, если указан порт или несколько портов, приложение производит проверку открытости этих портов;
8. Формат отчета по каждому серверу из входных данных должен содержать значение RTT (round trip time) в мс (ms), номер порта (при указании) и статус порта (opened или unknown). *Оговорка: если порт открыт это определяется однозначно, однако если порт не открыт, то это не означает, что он закрыт, он может не отвечать на запрос определенных протоколов с помощью которых определяется его статус.*

Требования к качеству кода

Производительность

Сайтов несколько (не более 10) частота опроса ограничена обменом данных по сети, поэтому дополнительные требования к производительности отсутствуют.

Масштабируемость

Каждый экземпляр приложения должен работать независимо. Необходимость в запуске нескольких экземпляров приложения на разных узлах сети может быть необходима для исключения проблем сетевой связности со стороны приложения. Например, если сначала пропадет сетевая связность у приложения, а затем у веб-сервера (на котором работает сайт), то администратор об этом не узнает.

Переносимость

Приложение должно функционировать в ОС Unix и ОС Windows.

Рекомендуемая версия Python: 3.10+

Доступность

24 часа 7 дней в неделю. Простои в работе означают отсутствие мониторинга. Убедиться в соблюдении данного требования возможно только в процессе опытной эксплуатации в течение определенного времени.

Надежность

В случае отсутствия доступа к сети или получения некорректных данных приложение должно выводить информацию об ошибке и продолжать работу.

Информационная безопасность

Требования к информационной безопасности отсутствуют (для упрощения разработки).

Качество при модернизации

Для каждого подключаемого модуля работы с сетью к приложению должна существовать возможность его замены (рекомендуется использовать паттерн «Адаптер»)

Требования к тестируемости кода отсутствуют (для упрощения разработки), однако рекомендуется создание модульных тестов для проверки валидности логики работы приложения.

Требования и рекомендации к коду

1. Для сокращения срока разработки не нужно использовать асинхронное программирование (Async IO), однако оно будет полезно при доработке решения;
2. У классов, их полей и методов должны быть простые и понятные имена, отражающие их суть. Также рекомендуется придерживаться PEP8 при разработке;
3. Классы должны быть распределены по модулям в соответствии с лучшими практиками (best practice) проектирования объектно-ориентированных приложений приложений (low in coupling, high in cohesion);

4. Для упрощения разработки модули журналирования (логи) и модуль передачи сообщений разрабатывать не нужно;
5. Рекомендуемые модули Python: **csv, socket, pythonping, contextlib, datetime**

Окончательный выбор используемых модулей остается за разработчиком.

Критерии оценки

Соответствие проекта требованиям к способности приложения (capabilities)

С помощью данной группы критериев будет дана оценка работоспособности приложения по заданным требованиям. (Работает ли вообще?)

1. Приложение осуществляет запросы согласно входным данным (могут изменяться при проверке работы приложения) и выводит информацию согласно требованиям.

Соответствие проекта требованиям к качеству качества при эксплуатации

С помощью данной группы критериев будет дана оценка выполнения требуемых характеристик качества при эксплуатации приложения. (Как работает?)

1. Работоспособность приложения при изменениях окружения (отсутствие доступа в сеть)
2. Работоспособность приложения при получении некорректных данных
3. Работоспособность приложения в ОС Unix и ОС Windows

Соответствие проекта требованиям качества при модернизации

С помощью данной группы критериев будет дана оценка выполнения требуемых характеристик качества при модернизации. (Насколько сложно сопровождать код? Насколько сложно расширять функционал приложения?)

1. Именованние классов/методов/полей/переменных;
2. Композиция классов, разделение ответственности между классами и модулями;
3. Рациональность выбора типов данных для хранения и передачи информации внутри приложения.

Каждому критерию будет дана оценка по пятибалльной шкале:

0 баллов – элемент отсутствует;

1 балл – элемент в наличии, есть критические ошибки;

2 балла – элемент реализован «удовлетворительно», возможно, есть некритические ошибки;

3 балла – элемент реализован «хорошо»;

4 балла – элемент реализован на «отлично»;

5 баллов - получает лучшая реализация элемента по данному критерию среди всех участников.

Заключительные слова

Данная задача обладает определенной образовательной ценностью. При желании испытуемый может продолжить над ней работу и после олимпиады, дополнив полезными возможностями, такими как: логирование, логика оповещений пользователя (TG, почта), асинхронные запросы к серверу, проверка корректности возвращаемой страницы сайта (модуль requests) и т. д.

Рекомендуемые материалы

- Онлайн-урок эксперта номинации: <https://youtu.be/o3LHY8bGjAs>
- Сведения по диагностике сетевых подключений (протокол ICMP, команда ping, работа DNS сервера) возможно найти в открытом доступе.
- Краткая статья по качеству ПО (интерпретация автора):
<https://codingness.ru/mind:quality>
- Официальная документация по подключаемым модулям:

<https://docs.python.org/3/>

Приложения

Пример файла CSV с входными данными

Файл составлен для проведения тестирования приложения по всем возможным сценариям использования (доменное имя с некорректными портами, доменное имя, отсутствие адреса или доменного имени, доменное имя и порт, доменное имя и несколько портов, IP-адрес и порт). Рекомендуется изменить IP-адрес на любой доступный во время тестов.

```
Host;Ports
ya.ru;xzx1
localhost;
;80
yandex.ru;443
last.fm;80,443
172.16.3.1;53
192.168.1.210;53
```

Пример вывода приложения

В примере вывода приложения ошибки в файле входных данных игнорируются


```
['ya.ru', ['87.250.250.242'], []]
```

```
2023-02-07 20:52:52.026665 | ya.ru | 87.250.250.242 | 0.0 | 4.46 ms | -1 | ???
```

```
['localhost', ['127.0.0.1'], []]
```

```
2023-02-07 20:52:52.071774 | localhost | 127.0.0.1 | 0.0 | 0.02 ms | -1 | ???
```

```
['yandex.ru', ['5.255.255.5', '5.255.255.60', '77.88.55.77', '77.88.55.80'], ['443']]
```

```
2023-02-07 20:52:52.121340 | yandex.ru | 5.255.255.5 | 0.0 | 4.91 ms | 443 | Opened
```

```
2023-02-07 20:52:52.186996 | yandex.ru | 5.255.255.60 | 0.0 | 8.29 ms | 443 | Opened
```

```
2023-02-07 20:52:52.288615 | yandex.ru | 77.88.55.77 | 0.0 | 12.83 ms | 443 | Opened
```

```
2023-02-07 20:52:52.379700 | yandex.ru | 77.88.55.80 | 0.0 | 8.31 ms | 443 | Opened
```

```
['last.fm', ['34.96.123.111'], ['80', '443']]
```

```
2023-02-07 20:52:52.550413 | last.fm | 34.96.123.111 | 0.0 | 16.2 ms | 80 | Opened
```

```
2023-02-07 20:52:52.567176 | last.fm | 34.96.123.111 | 0.0 | 16.2 ms | 443 | Opened
```

```
['???', ['172.16.3.1'], ['53']]
```

```
2023-02-07 20:52:52.607987 | ??? | 172.16.3.1 | 0.0 | 7.76 ms | 53 | Opened
```

```
['???', ['192.168.1.210'], ['53']]
```

```
2023-02-07 20:53:12.651287 | ??? | 192.168.1.210 | 1.0 | 2000 ms | 53 | Not opened
```

Пояснение:

В приложении дан пример вывода, который не в точности соответствует тому, что должно получиться, однако очень близок к нему. В примере 0 или другое число после IP-адреса означает число потерянных пакетов, т.к. тестирование производится по одному ICMP запросу в этом поле будет или 0 или 1. В задании нет требования по выводу данных о потерях пакетов, поэтому можно это не учитывать.

Для понимания: Вот один из вариантов кода, формирующий строку вывода:

```
def show(self) :  
  
    result = self.check_date_and_time.strftime("%Y-%m-%d %H:%M:%S.%f")  
  
    result += ' | ' + self.hostname  
  
    result += ' | ' + self.ip_address  
  
    result += ' | ' + str(self.packet_loss)  
  
    result += ' | ' + str(self.rtt_max_ms) + ' ms'  
  
    result += ' | ' + str(self.port)  
  
    result += ' | ' + self.port_check_state  
  
    print(result)  
  
    pass
```

Плагиат и несамостоятельное выполнение

Работы, уличенные в плагиате, оцениваться не будут: участники, приславшие не оригинальные работы — из дальнейшего участия выбывают.

Участие индивидуальное. Командное участие не допускается.

Оргкомитет оставляет за собой право связаться выборочно с любым из участников полуфинала перед подведением итогов, чтобы убедиться в самостоятельном выполнении задания.

Определение финалистов

Итоги подведем **не позднее 31 марта**. Результаты полуфинала опубликуем на сайте Олимпиады в виде рейтинговых таблиц. Финалистами станут 15 человек, показавшие лучшие результаты. При определении финалистов будут учтены условия допуска (приложение 4 [Положения об олимпиаде](#)).