

ИНТЕРАКТИВНЫЕ ДИСПЛЕИ ДЛЯ УЧЕБНЫХ АУДИТОРИЙ

Чайбар А.А.

Научный руководитель: Фадеев А.С.

Томский политехнический университет, Институт кибернетики

arty1901@gmail.com

Введение

Сегодня расписание занятий студентов ТПУ предоставлено в виде веб-списка, доступ к которому возможен с помощью мобильного приложения «расписание вузов», с мобильного или стационарного компьютера, или электронного терминала, установленного в рекреациях университета.

Для проверки текущего расписания ТПУ необходимо запустить программу-браузер или мобильное приложение и иметь подключение к Интернету. Помимо этих требований, эта процедура занимает определенное время, которое при опоздании студента становится важным.

Основной задачей данной работы является исключить необходимость выполнения процедуры поиска нужного расписания в определенной аудитории. Предполагаемое решение поставленной задачи — отображение текущего расписания на интерактивном дисплее, который должен быть установлен возле каждой учебной аудитории.

Цель работы

Целью работы является – разработка информационной системы, отображающей информацию о текущем занятии в конкретной аудитории в режиме реального времени с уточнением наименования учебной дисциплины, имени преподавателя и номеров группы, для которых занятие проводится. Так же, для дополнительной информативности, должна отображаться информация о следующей и предыдущей паре. Такой способ информирования облегчит ориентирование студентов младших курсов в расписании.

Для реализации технической части было принято решение использования встраиваемые мини-компьютеры «Raspberry Pi» и дисплей диагональю 5-7”. Программная часть, реализующая получение, формирование и вывод расписания, реализуется на языке программирования PHP, с применением веб-технологий HTML и CSS.

Архитектура информационной системы

При выборе архитектуры построения системы, было принято решение отказаться от разработки приложений для мини-компьютеров Raspberry Pi, что позволит использовать их в качестве сверхтонких клиентов, представляющих собой простой веб-браузер. Так как серверная часть (gasp.tpu.ru) не предоставляет требуемые сервисы,

появилась необходимость использования трехзвенную архитектуру, состоящую из семейства мини-компьютеров Raspberry Pi, промежуточного сервера и сервера расписания ТПУ. При установке на большом количестве аудиторий территориальная разбросанность не позволит оперативно вносить изменения в код программы на тонких клиентах, что существенно усложнит их администрирование и настройку. При наличии на конечных клиентах только программы-браузера, необходимо настраивать только параметры подключения к сети и запрос к серверу с идентификатором определенной аудитории. Все остальные программные функции будут выполняться на одном промежуточном сервере. На схеме (рисунок 1) показана работа всей системы от формирования запроса до вывода информации на интерактивный дисплей аудитории.

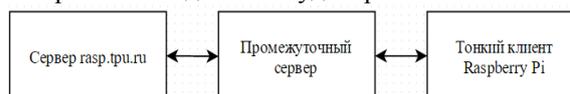


Рисунок 1 - Реализация запроса и ее вывод

В основе реализации программной части, лежит метод декомпозиции содержимого страницы, получаемой от сервера расписаний ТПУ, включающий в себя последовательный синтаксический анализ информации, размещенной на странице сайта.

Изначальный запрос формируется в веб-браузере тонкого клиента и содержит лишь идентификатор той аудитории, где он установлен. Этот запрос обрабатывается промежуточным сервером и на его основе формируются запросы к серверу расписаний ТПУ. Нужно отметить, что с точки зрения сервера gasp.tpu.ru, промежуточный сервер является обычным клиентом. Ответ с сервера gasp.tpu.ru возвращается в промежуточный сервер, где на веб-сервере формируется веб-страница для передачи в ответ тонкого клиента. Полученная страница передается на устройство Raspberry Pi и отображается в неизменном виде.

Формат веб-страницы

Информация, выводимая на дисплей (представляющий собой табличку-указатель конкретной аудитории) представляет собой веб-страницу, которая содержит информацию, представленную несколькими блоками, как о данной аудитории, так и о занятиях, проводимых в ней.

Первым блоком информации является номер аудитории и номер учебного корпуса. Следующий блок содержит значение текущего времени.

Далее содержится 3 блока: первый блок информации о **предыдущей паре**, второй блок – **текущая пара**, третий блок – **следующая пара**. Каждый из этих блоков содержит следующие атрибуты: *название дисциплины, имя преподавателя, номер группы, тип аудиторного занятия*. Так же имеется дополнительный атрибут - *примечание* - в блоке текущей пары.

Программная реализация

Для реализации и вывода информации в виде веб-страницы, на промежуточном сервере размещены файлы: `subject.php`, `time.php`, `name_class.php`, `group.php`, которые являются ключевой программной составляющей.

Так как реализация данных файлов является общей, то основные этапы написания программы являются общими. Поставленные этапы реализации:

1. получение содержимого с сайта расписания;
2. выделение требуемых объектов;
3. сортировка;
4. формирование результата в виде веб-страницы.

Реализация первой задачи, осуществляется функцией `file_get_contents()`.

В переменную `$url` задается URL-ссылка на веб-страницу. Все содержимое передается в переменную `$content` через функцию `file_get_content()`. Результатом работы функции на данном этапе является вся веб-страница данного адреса.

Следующий этап – выделение требуемой информации.

Написания данной части кода основывается на поиске в найденном. На этом этапе были использованы функции `preg_match()` и `preg_match_all()`.

Для написания шаблонов были использованы регулярные выражения. Шаблон состоит из следующих составляющих: **ограничителя шаблона, самого шаблона** и не обязательных **модификаторов**:

`preg_match('#<div class="j-spoiler-element j-spoiler-element-visible">(.*?)</div>#Uis', $content, $array)`.

Как показано выше, шаблоном для поиска является текст, находящийся между ограничителями решетка «#». Функция `preg_match()` осуществляет поиск всех блоков `div` с указанным классом. Так же используется подшаблон `(.*)`, который ищет любое совпадение между тегами `div`. Результатом второго этапа является не отсортированный массив повторяющихся элементов, с требуемым объектом поиска.

Третий этап – сортировка. Для избавления от повторений в массиве, используется функция `array_unique()` – убирает все повторяющиеся значения из массива, но сохраняет их ключи без

изменений. Для сортировки ключей элементов полученного массива, после функции `array_unique()`, используется функция `array_values()`. Она заново индексирует возвращаемый массив числовыми индексами.

После первых 3-х этапов, результатом выполнения файлов будут массивы соответствующих значений: время начала каждой пары, названия предметов, ФИО преподавателей, тип аудиторного занятия, номера групп. Четвертый этап – вывод результата.

Для вывода конечного результата был написан шаблон `index.php`. Файл формирует небольшую веб-страницу на языке HTML, к которой подключаются вышеперечисленные PHP-файлы. Написанная PHP-программа, загружается в отдельно выделенный стационарный промежуточный веб-сервер.

Для оформления начертания шрифтов, цветового решения и дизайна веб-страницы, применено оформление с использованием каскадных таблиц стилей CSS.

Заключение

В настоящий момент ведется тестовая эксплуатация разработанных компонентов, продолжается работа над созданием легко читаемого и красочного оформления интерактивного указателя, представленного веб-страницей, и адаптация содержимого к маленьким дисплеям (диагональ которых составляет 7 дюймов) с низким разрешением.

В качестве следующего этапа планируется разработка единого технического решения, позволяющего легко тиражировать интерактивные указатели в виде законченных отлаженных готовых устройств.

Список использованной литературы

1. Парсинг html-сайтов с помощью PHP, Ruby, Python. [Электронный ресурс]. – URL: <http://parsing.valemak.com/> (дата обращения: 22.10.2015).
2. Регулярные выражения для пыхтящих. [Электронный ресурс]. – URL: <http://pyha.ru/forum/topic/19> (дата обращения: 22.10.2015).
3. Справочник по PHP. [Электронный ресурс]. – URL: <http://php.net/> (дата обращения: 22.10.2015).
4. Справочник по PHP. [Электронный ресурс]. – URL: <http://www.php.su/> (дата обращения: 22.10.2015).
5. Модификаторы регулярных выражений. [Электронный ресурс]. – URL: <http://sbfactory.ru/cd/?p=2334> (дата обращения: 22.10.2015).