OneNote同步服务器设计方案

# 程序执行图



# OneNote大量数据传输

1. Linux服务器提供一个守护进程；
2. 客户端访问上载URL时：
   1. PHP（Apache中执行的脚本）对客户端进行认证，如果认证通过并生成一32位的随机编码；
   2. PHP脚本将该编码以及编码生成时间插入到上载的数据库表中，并将该编码返回给客户端；
   3. 客户端收到上载编码后则启用套接字连接服务器，并将编码传输给服务器；
   4. 服务器守护进程将收到的编码与数据库中的记录进行对比，如果编码可使用并且与生成时间之差未超过3min，则接收客户端上载数据；
   5. 服务器守护进程将接收到的数据插入到数据库中，结束数据上载。
3. 客户端访问下载URL时：
   1. PHP（Apache中执行的脚本）对客户端进行认证，如果认证通过并生成一32位的随机编码；
   2. PHP脚本将该编码以及编码生成时间插入到下载的数据库表中，并将该编码返回给客户端；
   3. 客户端收到上载编码后则启用套接字连接服务器，并将编码传输给服务器；
   4. 服务器守护进程将收到的编码与数据库中的记录进行对比，如果编码可使用并且与生成时间之差未超过3min，则接受客户端下载数据要求；
   5. 服务器守护进程将数据从数据库中取出通过套接字传给客户端，结束数据下载。

# 守护进程需求及流程

守护进程接收到客户端的连接后：

1. 客户端发送（32+1+4）字节数据给服务端，其中32字节是URL生成的随机编码，1字节是获取数据类型（1：上载数据；0：下载暑假），如果是上载数据，则4字节代表上载数据的大小，如果是下载数据该4字节保留。
2. 服务端获取到客户端数据后，将判断32位编码是否正确、数据操作方式是否与记录一直，正确则返回32字节暑假，即是(32+4)字节数据，前32位代表编码，如果上载数据，则4字节为0000，下载数据则是下载数据大小；拒绝要求则返回36个字节的0。
3. 客户端接收到服务端发送的32位编码，如果服务端接受请请：
   1. 如果是上载数据，则将数据上载，上载完后等待服务端返回，需等待服务端再次返回32位编码，数据上载才完成；
   2. 如果是数据下载，客户端接收到数据编码后等待接收数据，服务端发送数据完成后等待客户端返回32位编码进行确认。
4. 断开客户端连接并结束线程。

守护进程：

每个OneNote的page都存在PageSettings、Title元素。

接收到整个XML数据后，将数据与数据库进行对比：

1. 检测PageSettings、Title是否有修改，有则刷新数据库数据；
2. 检测outline内容：

outline是page页面的核心数据，一个page可以存在多个outline，而每个outline有且仅有一个OEChildren元素，OEChildren元素又由多个OE，对比时候：

1. 检测是否有新的OE（通过ID来判断），有则需更增加数据库数据；
2. 检测已经存在OE的OElastModifiedTime属性，如果时间比数据库更新，则徐更新数据库数据。

# windows客户端

windows客户端使用C#编写，主要使用COM组件将OneNote页面内容进行提取、删除、插入等操作。