

上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

课程报告



mini-PACS 系统的搭建

518021910971 裴奕博

目录

1	开源软件的选择	2
1.1	SCP 的选择	2
1.2	SCU 的选择	2
2	mini-PACS 系统的搭建	3
2.1	Server 的配置	3
2.2	通过 Web 应用上传 Dicom 文件	3
2.3	通过 DCMTK 上传 Dicom 文件	4
2.4	检索和查看 Dicom 文件	4
3	log 日志内容解析	5
3.1	log 日志获取	5
3.2	log 日志解析	5
3.2.1	Associate RQ/AC PDU 的交换	5
3.2.2	DIMSE 的发送和接收	6
3.2.3	数据在 SCP 上的存储	8
4	实验感想	8

摘要

中文摘要 **Keywords:** 关键字

1 开源软件的选择

1.1 SCP 的选择

在上次的开源软件调研报告中，我了解到了几个目前流行的开源 mini-PACS 框架，其中就有 Orthanc。Orthanc 是一个基于 C++ 开发的轻量级 mini-PACS 框架，采用 GPL 协议开源。它具有以下几个特点 [1]:

1. 提供了内置的 http 服务，可以通过 Web 应用来可视化的上传、查看、检索和传输 Dicom 文件。
2. 跨平台。可以在 Windows 下运行，也可以通过 Docker 在 Linux 下运行，可以部署在常见的网站服务器上如 Nginx、Apache 等。
3. 可以通过多种方式实现 Dicom 的传输。不仅提供了 Dicom Server 端口，也可以使用 Web 应用可视化上传，更是提供了 REST API 直接上传 Dicom 文件。

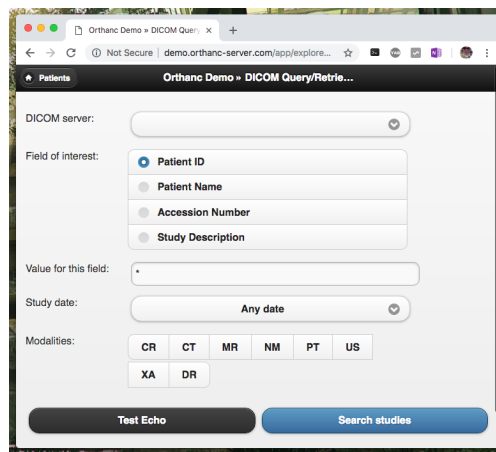


图 1: Orthanc 界面

Orthanc 目前仍然在持续更新中，更加现代化，界面也漂亮舒适。因此在本次实验中选择了 Orthanc 作为 Dicom 服务器。

1.2 SCU 的选择

由于 Orthanc 提供了多种传输方式，因此此次实验既尝试使用了开源的 DCMTK 作为 SCU，也尝试了直接使用 Web 应用上传。

DCMTK 是由德国 offis 公司提供的开源项目, 使用 C++ 实现了 Dicom 协议的相关细节, 并封装成了各个应用, 为我们提供了实现 DICOM 协议的一个平台, 使得我们可以在它的基础上轻松的完成自己的主要工作, 而不必把太多的精力放在实现 DICOM 协议的细节问题上 [2]。实际上, Orthanc 的后端也集成了 DCMTK 工具包用于 Dicom 文件的相关操作。

2 mini-PACS 系统的搭建

2.1 Server 的配置

根据 Orthanc 的官方文档 [3], Orthanc 在安装完毕时即提供了默认配置, 包括默认的 sqlite 数据库、默认的 Dicom 服务、默认的 http 服务及其端口, 因此只需要修改自己需要的部分, 其他部分均由 Orthanc 读取默认配置文件。本次实验采用的配置文件 server_config.json 内容如下:

```
1 {  
2     "Name": "My archive",  
3     "HttpPort": 4200,  
4     "DicomAet": "ARCHIVE",  
5     "DicomPort": 104  
6 }
```

由于使用了自定义的配置文件, 因此启动 Orthanc 时需要指定配置文件路径。因此使用了 bat 脚本来方便 Server 的启动。脚本 runOrthanc.bat 内容如下:

```
1 "D:\Program Files\Orthanc Server\Orthanc" ./server_config.json --trace
```

参数 -trace 或 -verbose 可以提供调试时的相关信息, 由于需要观察日志, 因此将 -trace 选项打开。

2.2 通过 Web 应用上传 Dicom 文件

通过运行上述脚本运行 Orthanc, 此时 http Server 端口为 4200, Dicom Server 端口为 104。进入浏览器, 输入 localhost:4200 进入 Orthanc 界面, 点击顶部的 Upload 菜单即可进入上传界面。

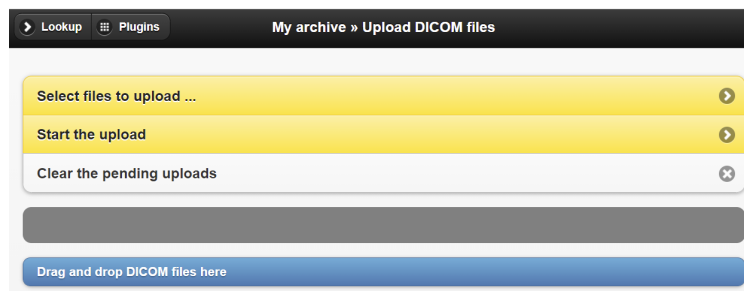


图 2: Orthanc 上传界面

在此界面即可通过拖动和选择文件的方式在图形界面中上传文件。

2.3 通过 DCMTK 上传 Dicom 文件

上述的方式虽然方便，但需要逐个选择上传的文件。如碰到数据量较大的情况就很不方便。此时可以选择通过 DCMTK 借助命令行脚本的方式上传。

在 DCMTK 中，C-STORE 的命令被封装在 store scu.exe 中，通过 DCMTK 官网的文档，我们可以得知其使用方法 [4]：

```
1 store scu.exe -d -aec ARCHIVE localhost 104 $file
```

命令行参数-d 输出调试信息，-aec 指定应用实体名字，并指定端口号和上传文件。通过 PowerShell 脚本 (storeDCM.ps1) 即可批量上传一个文件夹下的所有 DCM 文件。足够实现一个序列或者检查的批量上传。

```
1 $files = Get-ChildItem -Path .\*.dcm
2 foreach( $file in $files )
3 {
4     store scu.exe -d -aec ARCHIVE localhost 104 $file
5 }
```

2.4 检索和查看 Dicom 文件

这一部分功能均由 Orthanc 内置。通过 Lookup 选项菜单即可查看所有 Dicom 文件信息。通过病人-检查-序列-实例的层级结构呈现，如下图 3所示。

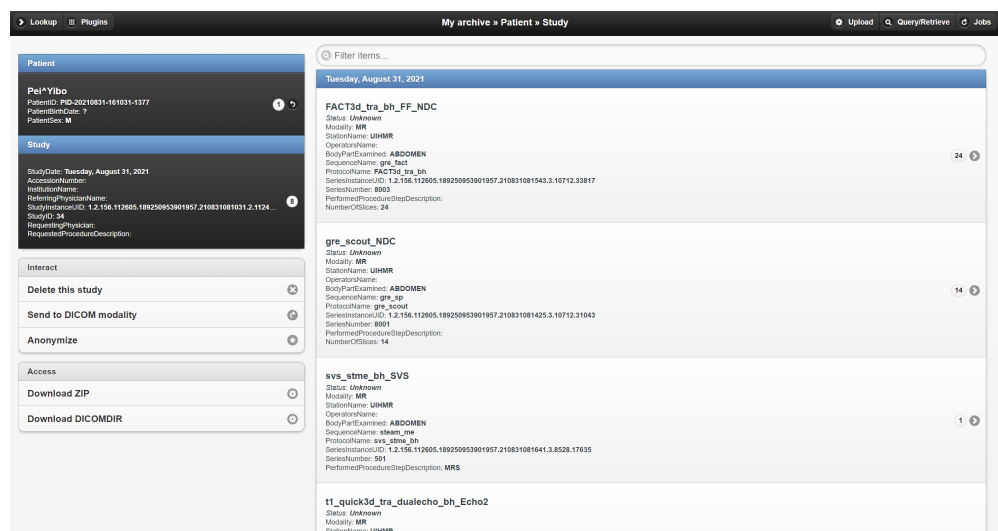


图 3: Orthanc 查看界面

3 log 日志内容解析

3.1 log 日志获取

在 VSCode 中，打开工作目录，启动 Orthanc，使用命令行上传单个 Dcm 文件，分成两个窗口同时查看 SCU 和 SCP 的日志信息 [5]：

图 4: log 日志获取

最终得到的日志结果见 client_log.log、server_log.log 和 client_fail_log.log 文本文件。

3.2 log 日志解析

表 1: 缩写和名称对照表

缩写	全称（英文）	全称（中文）
PDU	Protocol Data Units	协议数据单元
DIMSE	DICOM Message Service Element	DICOM 消息服务
SOP	Service Object Pair	服务对象对

无论从 SCU 还是 SCP 来看，日志可以分为以下几个部分：

1. Associate RQ/AC PDU 的交换
2. DIMSE 的发送和接收
3. 数据在 SCP 上的存储

3.2.1 Associate RQ/AC PDU 的交换

这一部分可以分为如下几个步骤：

1. SCU 生成并发送 Associate RQ PDU。
2. SCP 收到 Associate RQ PDU 后根据状态返回 Associate AC PDU（接受连接）或 Associate RJ PDU（拒绝连接）。

Associate AC PDU 中包含如下信息：

```
===== BEGIN A-ASSOCIATE-AC =====
Our Implementation Class UID:      1.2.276.0.7230010.3.0.3.6.6
Our Implementation Version Name:   OFFIS_DCMTK_366
Their Implementation Class UID:    1.2.276.0.7230010.3.0.3.6.6
Their Implementation Version Name: OFFIS_DCMTK_366
Application Context Name:          1.2.840.10008.3.1.1.1
Calling Application Name:          STORESCU
Called Application Name:           ARCHIVE
Responding Application Name:       ARCHIVE
Our Max PDU Receive Size:         16384
Their Max PDU Receive Size:       16384
Presentation Contexts:
  Context ID:      1 (Accepted)
    Abstract Syntax: =AmbulatoryECGWaveformStorage
    Proposed SCP/SCU Role: Default
    Accepted SCP/SCU Role: Default
    Accepted Transfer Syntax: =LittleEndianExplicit
  Context ID:      3 (Accepted)
    Abstract Syntax: =AmbulatoryECGWaveformStorage
    Proposed SCP/SCU Role: Default
    Accepted SCP/SCU Role: Default
    Accepted Transfer Syntax: =LittleEndianImplicit

*****
Very Long Duplicate Parts
*****

  Context ID:      255 (Accepted)
    Abstract Syntax: =XRayRadiofluoroscopicImageStorage
    Proposed SCP/SCU Role: Default
    Accepted SCP/SCU Role: Default
    Accepted Transfer Syntax: =LittleEndianImplicit
Requested Extended Negotiation: none
Accepted Extended Negotiation: none
Requested User Identity Negotiation: none
User Identity Negotiation Response: none
===== END A-ASSOCIATE-AC =====
```

图 5: Associate AC PDU

其中包括协议的 class ID，协议实现的版本号，PDU 的大小，应用上下文 Context 等信息。Context 共有 256 段，每段都会返回连接状态（Accepted/Proposed）、Abstract Syntax、SCP/SCU Role 等信息。这一部分的信息不包括上传文件的信息，仅供建立连接使用。

3.2.2 DIMSE 的发送和接收

这一部分可以分为如下几个步骤：

1. SCU 以 OUTGOING DIMSE MESSAGE 形式发送一个 C-STORE RQ。
2. SCP 收到 INGOING DIMSE MESSAGE 后根据状态同样以 OUTGOING DIMSE MESSAGE 形式发送 C-STORE RSP 给 SCU。
3. SCU 收到响应后确认状态，整个 STORE 流程结束。

这一过程中客户端和服务端的 log 显示如下：

```
I: Sending Store Request (MsgID 1, MR)
D: ===== OUTGOING DIMSE MESSAGE =====
D: Message Type      : C-STORE RQ
D: Message ID       : 1
D: Affected SOP Class UID : MRImageStorage
D: Affected SOP Instance UID : 1.2.156.112605.189250953901957.210831081543.4.10712.144023
D: Data Set         : present
D: Priority          : medium
D: ===== END DIMSE MESSAGE =====
D: DcmDataset::read() TransferSyntax="Little Endian Implicit"
I: Received Store Response
D: ===== INCOMING DIMSE MESSAGE =====
D: Message Type      : C-STORE RSP
D: Presentation Context ID : 113
D: Message ID Being Responded To : 1
D: Affected SOP Class UID : MRImageStorage
D: Affected SOP Instance UID : 1.2.156.112605.189250953901957.210831081543.4.10712.144023
D: Data Set         : none
D: DIMSE Status      : 0x0000: Success
D: ===== END DIMSE MESSAGE =====
I: Releasing Association
```

图 6: DIMSE Client Log

```
===== INCOMING DIMSE MESSAGE =====
Message Type      : C-STORE RQ
Presentation Context ID : 113
Message ID       : 1
Affected SOP Class UID : MRImageStorage
Affected SOP Instance UID : 1.2.156.112605.189250953901957.210831081543.4.10712.144023
Data Set         : present
Priority          : medium
===== END DIMSE MESSAGE =====
```

图 7: DIMSE Server Log

可以清楚的看到 SCU 和 SCP 通信时信息的发送和接收。

C-STORE RQ 的内容包括以下几个字段：消息类型、消息 ID、SOP 的类和实例 UID、数据集合消息优先级。C-STORE RSP 的内容包括以下几个字段：消息类型、上下文 ID、响应对象的 ID、SOP 的类和实例 UID、数据集和 DIMSE 状态等信息。

最后 DIMSE 状态若显示为 success，则整个 DICOM 网络通信流程成功进行。

3.2.3 数据在 SCP 上的存储

```
11006 11:12:33.454129 main.cpp:352] Incoming Store request from AET STORESCU on IP 127.0.0.1, calling AET ARCHIVE
11006 11:12:33.460128 FilesystemStorage.cpp:124] Creating attachment "2c07079e-adf0-472d-b240, calling AET ARCHIVE 0-e887f23bf93f"
of "DICOM" type (size: 1MB)
11006 11:12:33.470652 FilesystemStorage.cpp:258] Deleting attachment "2c07079e-adf0-472d-b240-e887f23bf93f" of type
0-e887f23bf93f of typ
11006 11:12:33.476661 ServerContext.cpp:628] Already stored
11006 11:12:33.484665 CommandDispatcher.cpp:917] (dicom) Finishing association with AET STORESCU on IP 127.0.0.1: DUL Peer
Requested Release
11006 11:12:33.489661 CommandDispatcher.cpp:925] (dicom) Association Release with AET STORESCU on IP 127.0.0.
1
CU on IP 127.0.0.1
11006 11:12:34.168000 Connection.cpp:403] (sqlite) SQLite::Connection::FlushToDisk
11006 11:12:41.939328 ServerContext.cpp:254] Serializing the content of the jobs engine
```

图 8: DB Store Log

在通信流程结束后, Orthanc 将上传的 Dcm 存储到数据库中(此处使用的是默认的 sqlite), 可在本地的 OrthancStorage 目录下找到存储的数据。

4 实验感想

本次实验尝试使用了现有的开源软件搭建了 mini-PACS 服务器, 并尝试了 Dicom 文件的上传并查看了日志。在实验过程中, 遇到过如下几个问题:

1. 端口冲突。在调试时, 有时会发现 Orthanc 由于端口冲突无法启动的情况。
解决方法: 在命令行 kill 掉占用端口的进程或者修改配置文件换用其他端口号。
2. 在尝试本地 WSL 中启动 Linux Docker 下的 Orthanc, 模拟实际的远程存储情况时遇到 HTTPS encryption is disabled 错误。
解决方法: 根据官网的提示重新认证证书或使用 Nginx, 但还未实际实现。

本实验所用的脚本和 log 文件均已附在压缩包中, 附上 Orthanc 下载地址: <https://www.orthanc-server.com/download.php>。

参考文献

- [1] Wikipedia contributors. Orthanc (server) — Wikipedia, the free encyclopedia, 2021. [Online; accessed 6-October-2021].
- [2] 百度百科. 百度百科 — dcmtk, 2021.
- [3] Osimis. Orthanc book, 2021.
- [4] Doxygen. Dcmtk documentation, 2021.
- [5] NEMA. Dicom standard: Current edition, 2021.