

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	Server 和 SCP 的选择	2
1.2	SCU 的选择	2
2	mini-PACS 系统的搭建	3
2.1	mini-PACS 系统概览	3
2.2	Server 的配置	3
2.3	SCP 的配置	4
2.4	通过 Web 应用上传 Dicom 文件	5
2.5	C-STORE 的实现	5
2.6	C-FIND 的实现	6
2.7	C-MOVE 的实现	6
3	log 日志内容解析	7
3.1	C-STORE DIMSE 内容解析	8
3.2	C-FIND DIMSE 内容解析	9
3.3	C-MOVE DIMSE 内容解析	11
4	实验感想	12

摘要

本报告详细叙述了一个 mini-PACS 系统的搭建流程，从开源软件的选择到搭建和功能的调试和 log 的解析，以及在搭建过程中遇到的各种问题。**关键字：**开源软件，mini-PACS，Orthanc，DCMTK

1 开源软件的选择

1.1 Server 和 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。

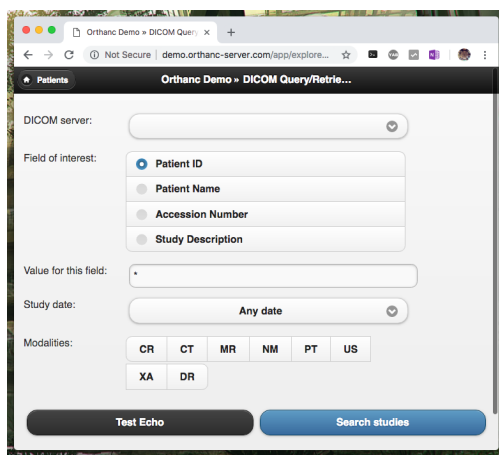


图 1: Orthanc 界面

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 Server 和 SCP。其中，Server 运行在 windows 本地，SCP 运行在 WSL 的 Docker 下。

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 mini-PACS 系统概览

本次实验使用的 mini-PACS 系统由如下三个部分组成：SCU、本地 Server 和运行在 Linux 上的模拟远程 SCP。他们依靠 Dicom 网络传输通信协议连接。整个系统的结构框图如下图 2：

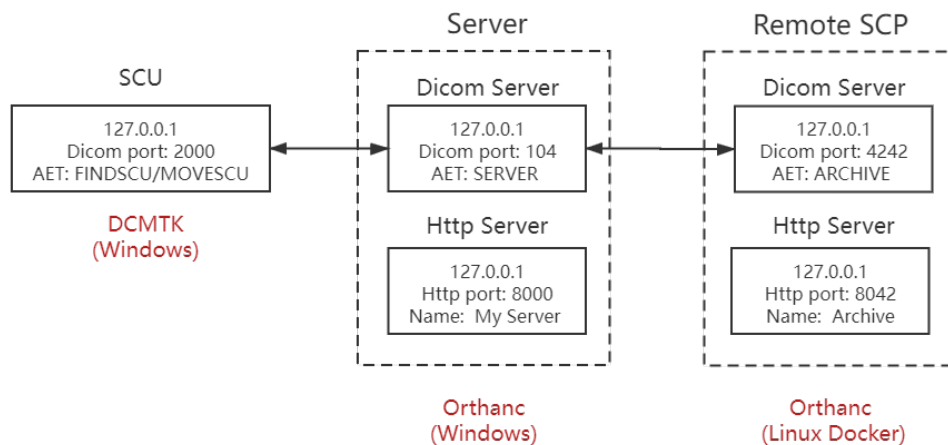


图 2: 整体系统框图

2.2 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，其他部分均会由 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 // server_config.json
2 {
3     "Name": "My Server",
4     "HttpPort": 8000,
5     "DicomAet": "SERVER",
6     "DicomPort": 104,
7     "RemoteAccessAllowed": true,
```

```
8     "DicomModalities" : {  
9         "local" : [ "FINDSCU", "127.0.0.1", 2000 ],  
10        "local2" : [ "MOVESCU", "127.0.0.1", 2000 ],  
11        "archive": [ "ARCHIVE", "127.0.0.1", 4242 ]  
12    }  
13 }
```

Server 运行在本地 104 端口下，同时启动了一个 4200 端口下的 http 服务。Dicom 网络中的其他几个部分也需要在此处声明用于网络传输的连接。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。

2.3 SCP 的配置

SCP 运行在 WSL 的 Docker 环境下，与 Server 的配置同理，根据官方文档 [3]，采用配置如下：

```
1 # docker-compose.yml  
2 version: '3.1' # Secrets are only available since this version of Docker Compose  
3 services :  
4 orthanc:  
5     image: jodogne/orthanc-plugins:1.9.7  
6     command: /run/secrets/ # Path to the configuration files (stored as secrets)  
7     ports:  
8         - 4242:4242  
9         - 8042:8042  
10    secrets :  
11        - orthanc.json  
12    environment:  
13        - ORTHANC_NAME=HelloWorld  
14    secrets :  
15    orthanc.json :  
16        file : orthanc.json
```

```
1 // orthanc.json  
2 {  
3     "Name" : "Archive",
```

```
4     "RemoteAccessAllowed" : true,  
5     "DicomAet": "ARCHIVE",  
6     "DicomModalities" : {  
7         "server" : ["SERVER","127.0.0.1",104]  
8     }  
9 }
```

SCP 运行在本地 4242 端口下，同时创建了一个 8042 端口的 http 服务。启动时只需要使用命令

```
1 $ docker-compose up
```

即可启动服务。

2.4 通过 Web 应用上传 Dicom 文件

通过运行上述脚本运行 Orthanc Server，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。

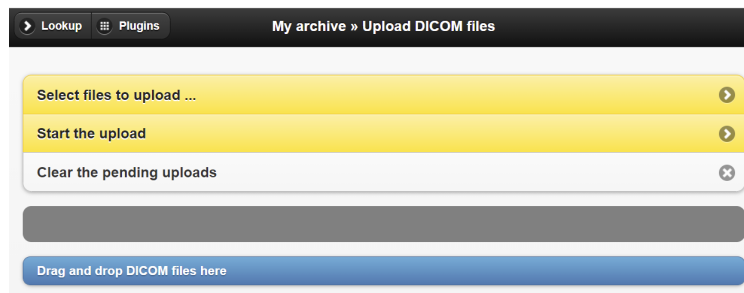


图 3: Orthanc 上传界面

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。

2.5 C-STORE 的实现

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     storescu.exe -d -aec SERVER localhost 104 $file
5 }
```

2.6 C-FIND 的实现

同样，C-FIND 也只涉及 SCU 和 Server 两个应用实体。同样通过 DCMTK 的官方文档得知，可以使用 findscu.exe 实现 C-FIND 功能，具体命令如下：

```
1 findscu.exe -ll trace -P localhost 104
2     -k QueryRetrieveLevel=Patient
3     -k PatientID=PID-20210831-161031-1377
4     > ./log/cfind_scu.log
```

根据以上条件可以查询到符合指定 PatientID 的所有 Dicom 文件。同时指定了 trace（最高的输出等级）并将 log 重定向至 cfind_scu.log 文件。

2.7 C-MOVE 的实现

与 C-STORE 和 C-FIND 不同，C-MOVE 涉及到三个应用实体，即 SCU、本地 Server 以及模拟的运行在 Linux 下的远程 SCP。同样通过 DCMTK 的官方文档得知，可以使用 movescu.exe 实现 C-MOVE 功能，具体命令如下：

```
1 movescu.exe -ll trace -aec SERVER localhost 104 -aem ARCHIVE
2     -k QueryRetrieveLevel=Patient
3     -k PatientID=PID-20210831-161031-1377
4     -k SeriesNumber=8003
5     > ./log/cmove_scu.log
```

可以将指定 PID 的指定序列从本地 Server 拷贝到 SCP。同时指定了 trace（最高的输出等级）并将 log 重定向至 cmove_scu.log 文件。

3 log 日志内容解析

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

每次运行时，都记录下 SCU 和 SCP 的日志，无论从 SCU 还是 SCP 来看，每次 Dicom 网络通信的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

而根据每次操作的种类和参数的不同，PDU 和 DIMSE 的内容会有所不同。

第一部分是 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 4: 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。

第二部分是 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 RQ。
2. SCP 收到 INGOING DIMSE MESSAGE 后根据状态同样以 OUTGOING DIMSE MESSAGE 形式发送 RSP 给 SCU。
3. SCU 收到响应后确认状态, 整个流程结束。

对于各种操作, Associate RQ/AC PDU 和 DIMSE 交换过程基本类似。不同的是后面的 DIMSE 的类型和参数不同, 因此以下分别对各自的 DIMSE 内容进行解析。

3.1 C-STORE DIMSE 内容解析

这一过程中的 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 5: C-STORE SCU DIMSE

```
===== 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 6: C-STORE SCP DIMSE

我们可以在其中看到两类 DIMSE，RQ 和 RSP。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 C-STORE 流程成功进行。

3.2 C-FIND DIMSE 内容解析

这一过程中的 log 显示如下：

```
D: ===== OUTGOING DIMSE MESSAGE =====
D: Message Type           : C-FIND RQ
D: Presentation Context ID : 1
D: Message ID             : 1
D: Affected SOP Class UID  : FINDPatientRootQueryRetrieveInformationModel
D: Data Set                : present
D: Priority                 : medium
D: ===== END DIMSE MESSAGE =====
I: Request Identifiers:
I:
I: # Dicom-Data-Set
I: # Used TransferSyntax: Little Endian Explicit
I: (0008,0052) CS [Patient] # 8, 1 QueryRetrieveLevel
I: (0010,0020) LO [PID-20210831-161031-1377] # 24, 1 PatientID
I:
D: DcmDataset::read() TransferSyntax="Little Endian Implicit"
D: DcmDataset::read() TransferSyntax="Little Endian Explicit"
I: Received Find Response 1
D: ===== INCOMING DIMSE MESSAGE =====
D: Message Type           : C-FIND RSP
D: Message ID Being Responded To : 1
D: Affected SOP Class UID  : FINDPatientRootQueryRetrieveInformationModel
D: Data Set                : present
D: DIMSE Status            : 0xff00: Pending: Matches are continuing
D: ===== END DIMSE MESSAGE =====
D: Response Identifiers:
D:
D: # Dicom-Data-Set
D: # Used TransferSyntax: Little Endian Explicit
D: (0008,0005) CS [ISO_IR 100] # 10, 1 SpecificCharacterSet
D: (0008,0052) CS [Patient ] # 8, 1 QueryRetrieveLevel
D: (0008,0054) AE [SERVER] # 6, 1 RetrieveAETitle
D: (0010,0020) LO [PID-20210831-161031-1377] # 24, 1 PatientID
D:
D: DcmDataset::read() TransferSyntax="Little Endian Implicit"
I: Received Final Find Response
D: ===== INCOMING DIMSE MESSAGE =====
D: Message Type           : C-FIND RSP
D: Message ID Being Responded To : 1
D: Affected SOP Class UID  : FINDPatientRootQueryRetrieveInformationModel
D: Data Set                : none
D: DIMSE Status            : 0x0000: Success: Matching is complete
D: ===== END DIMSE MESSAGE =====
```

图 7: C-FIND SCU DIMSE

```
===== INCOMING DIMSE MESSAGE =====
Message Type           : C-FIND RQ
Presentation Context ID : 1
Message ID             : 1
Affected SOP Class UID  : FINDPatientRootQueryRetrieveInformationModel
Data Set                : present
Priority                 : medium
===== END DIMSE MESSAGE =====
11010 10:28:06.945924 main.cpp:53] Incoming Find request from AET FINOSCU on IP 127.0.0.1, calling AET ANY-SCP
11010 10:28:06.949924 main.cpp:232] No limit on the number of C-FIND results at the Patient, Study and Series levels
11010 10:28:06.954924 main.cpp:242] No limit on the number of C-FIND results at the Instance level
11010 10:28:06.957924 FindScp.cpp:215] (dicom) Received C-FIND Request:

# Dicom-Data-Set
# Used TransferSyntax: Little Endian Explicit
(0008,0052) CS [Patient] # 8, 1 QueryRetrieveLevel
(0010,0020) LO [PID-20210831-161031-1377] # 24, 1 PatientID

11010 10:28:06.968926 OrthancFindRequestHandler.cpp:664] (dicom) DICOM C-Find request at level: Patient
11010 10:28:06.972925 OrthancFindRequestHandler.cpp:670] (dicom) (0008,0052) QueryRetrieveLevel = Patient
11010 10:28:06.975924 OrthancFindRequestHandler.cpp:670] (dicom) (0010,0020) PatientID = PID-20210831-161031-1377
11010 10:28:06.980926 ServerContext.cpp:1310] Number of candidate resources after fast DB filtering on main DICOM tags: 1
11010 10:28:06.984923 ServerContext.cpp:1435] Number of matching resources: 1
11010 10:28:06.987924 FindScp.cpp:326] (dicom) Sending C-FIND Response 1/1:

# Dicom-Data-Set
# Used TransferSyntax: Little Endian Explicit
(0008,0005) CS [ISO_IR 100] # 10, 1 SpecificCharacterSet
(0008,0052) CS [Patient] # 8, 1 QueryRetrieveLevel
(0008,0054) AE [SERVER] # 6, 1 RetrieveAETitle
(0010,0020) LO [PID-20210831-161031-1377] # 24, 1 PatientID

11010 10:28:07.006928 CommandDispatcher.cpp:917] (dicom) Finishing association with AET FINOSCU on IP 127.0.0.1: DUL Peer Requested Release
11010 10:28:07.011924 CommandDispatcher.cpp:925] (dicom) Association Release with AET FINOSCU on IP 127.0.0.1
```

图 8: C-FIND SCP DIMSE

我们可以在其中看到两类 DIMSE, RQ 和 RSP。C-FIND RQ 的内容包括以下几个字段: 消息类型、消息 ID、SOP 的类 UID、数据集和消息优先级。其中 SOP 类分为两种, FINDPatientRootQueryRe-

trieveInformationModel 和 FINDStudyRootQueryRetrieveInformationModel, 分别对应了 DICOM 标准中的两种数据获取模型。可以在使用 C-FIND 时根据需要指定参数切换。C-FIND RSP 的内容包括以下几个字段: 消息类型、响应对象的 ID、SOP 的类和实例 UID、数据集和 DIMSE 状态等信息。当最后 DIMSE 状态若显示为 success, 则整个 C-STORE 流程成功进行。

3.3 C-MOVE DIMSE 内容解析

对于 C-MOVE, 情况则有些不同。由于 C-MOVE 可能涉及多个文件的移动, 因此 SCU 会发送一次 RQ, 随后根据需要移动的文件数量连续返回若干个 RSP。显示的 log 如下图:

```
I: Sending Move Request
D: ===== OUTGOING DIMSE MESSAGE =====
D: Message Type           : C-MOVE RQ
D: Presentation Context ID : 3
D: Message ID             : 1
D: Affected SOP Class UID :
MOVEPatientRootQueryRetrieveInformationModel
D: Data Set                : present
D: Priority                 : medium
D: Move Destination        : ARCHIVE
D: ===== END DIMSE MESSAGE =====
```

图 9: C-MOVE RQ DIMSE

```
I: Received Move Response 2
D: ===== INCOMING DIMSE MESSAGE =====
D: Message Type           : C-MOVE RSP
D: Message ID Being Responded To : 1
D: Affected SOP Class UID :
MOVEPatientRootQueryRetrieveInformationModel
D: Remaining Suboperations : 261
D: Completed Suboperations : 2
D: Failed Suboperations    : 0
D: Warning Suboperations   : 0
D: Data Set                : none
D: DIMSE Status            : 0xff00: Pending: Sub-operations are continuing
D: ===== END DIMSE MESSAGE =====
```

图 10: C-MOVE RSP DIMSE

其中, RQ 只发送一次, 内容包括包括以下几个字段: 消息类型、消息 ID、上下文 ID、SOP 的类 UID、数据集、消息优先级和 C-MOVE 的目的地。其中 SOP 类与 C-FIND 对应分为两种, MOVEPatientRootQueryRetrieveInformationModel 和 MOVEStudyRootQueryRetrieveInformationModel, 分别对应了 DICOM 标准中的两种数据获取模型。可以在使用 C-MOVE 时根据需要指定参数切换。

返回的 RSP 与需要移动的文件数相同, 内容包括以下几个字段: 消息类型、响应对象的 ID、SOP 的类 UID、剩余、已完成、失败、警告的操作数量、数据集和 DIMSE 状态等信息。其中 DIMSE 状态可能包括: Success (全部完成), Sub-operations complete-No failures or warnings (单个文件完成), Sub-operations are continuing (正在移动当前文件) 等。当最后传输成功完成时, 会返回 0x0000: Success:

Sub-operations complete - No failures or warnings。

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. 在尝试三个 AET 时，发现 Dicom 网络通信无法连接。
解决方法：在各自的配置文件中添加所需的 AET 使其能够相互识别和通信。

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.