

电子科技大学
计算机科学与工程学院

标准实验报告

(实验) 课程名称 软件开发综合实验

电子科技大学教务处制表

电 子 科 技 大 学

实 验 报 告

小组：1 班 4 组	詹秋阳（2020060902021）、梁宇哲（2020070902021）、冯韵任（2020080903023）	
组长姓名：詹秋阳		学号：2020060902021
组员姓名：梁宇哲		学号：2020070902021
组员姓名：冯韵任		学号：2020080903023
一、实验室名称：基础实验大楼 510		
二、实验项目名称：文件备份软件		
<p>三、实验目的：</p> <p>本课程的教学目标是为学生提供理论与实践相结合的基础平台，以计算机科学与技术专业必修课程《程序设计（C 与 C++）》、《数据结构与算法》和《软件工程》三门课程所涉及的主要知识为核心，基于具体的工程项目，加深学生对理论知识的理解，使学生能够解决计算机系统的复杂工程问题，达到选择或设计恰当的逻辑结构、存储结构及相应的算法的培养目标，培养学生在实际软件生产开发过程中对其所学知识的综合运用能力，提升学生的计算思维能力、算法设计与分析能力、程序设计与编程实现能力、团队协作能力、计算机系统的认知、分析、设计和运用能力，工程实践能力。</p>		

四、实验内容：

实验内容：

1. 设计并实现一款数据备份软件，以项目组形式推进，每组最多三人。
2. 基于软件工程方法学进行项目推进，经历从需求分析、系统设计、编码实现、软件测试的整个软件生命周期。
3. 实验最终成果包括一款基本可用的软件及其对应文档。
4. 软件应包括指明的完整功能，重点考察其正确性、易用性、健壮性。
5. 软件文档应包括：需求分析说明书、系统设计文档、软件测试报告，重点考察其规范性、一致性、可读性。
6. 采用现代化软件开发工具辅助项目开发，包括但不限于：项目管理工具，UML 建模工具，集成开发环境，版本控制工具，软件测试工具。

该实验包括必选的基本要求和可任意选择的扩展要求：

基本要求：

各小组“独立”实现一款数据备份软件（对应基础分总分 40 分）：

数据备份：将目录树中的文件保存到指定位置

数据还原：将目录树中的文件恢复到指定位置

扩展要求：

各项目组根据自身情况自行选择扩展要求（对应扩展分总分）。

文件类型支持（10 分）：支持特定文件系统的特殊文件（管道/硬链接/软连接）

元数据支持（10 分）：支持对特定文件系统的文件元数据还原（属主/时间/权限等）

压缩解压（10 分）：通过文件压缩节省备份文件的存储空间

打包解包（10 分）：将所有备份文件拼接为一个文件保存

自定义备份（各 5 分）：允许用户筛选需要备份的文件（路径/类型/名字/时间/尺寸）

加密解密（10 分）：由用户指定密码，将所有备份文件均加密保存（库实现 7 分）

定时备份（10 分）：允许用户进行设置，周期性定时备份和数据淘汰

实时备份（15 分）：自动感知用户文件变化，进行自动备份

网盘备份（30 分）：将数据备份软件从单机模式扩展为网盘模式（10 分），还涉及到的功能包括：用户管理（5 分）、元数据管理（5 分）、传输加密（5 分）、增量备份（5 分）等。

其它功能：视功能难度讨论加分。

五、实验器材（设备、元器件）：

PC 多台，可连接互联网。

六、实验步骤及操作：

需求分析说明书（10 分）

1. 任务概述

1.1 引言

随着互联网的迅速发展，用户产生的数据量也不断增长，而威胁数据安全的事件却时有发生。用户数据一旦被破坏，恢复的成本相当之高，而唯一可以将损失降至最小且行之有效的方法便是数据备份。传统数据备份只能防止人为故障，且恢复时间较长。利用计算机信息技术开发数据备份软件，可以有效避免数据破坏，同时节省恢复数据的时间与成本。

该需求分析说明书根据数据备份软件的特点，完整描述了数据备份软件所要完成的工作，为软件开发者进行详细设计和编程提供基础，也为软件测试人员提供测试和验收的依据。通过该需求分析说明书可以全面了解数据备份软件所要完成的任务和所能达到的功能。

1.2 综合描述

1.2.1 产品的状况

本软件产品是电子科技大学《软件开发综合实验》的课程要求项目，旨在根据所学知识设计并实现一款数据备份软件。该数据备份软件独立于其他系统，应用方便。

1.2.2 产品的功能

数据备份软件的主要功能如下：

数据备份	将目录树中的文件保存到指定位置。
数据还原	将目录树中的文件恢复到指定位置。
文件类型支持	支持特定文件系统的特殊文件（管道/软链接/硬链接等）。
元数据支持	支持特定文件系统的文件元数据（属主/时间/权限等）。
打包解包	将所有备份文件拼接为一个大文件保存。
加密解密	由用户指定密码，将所有备份文件均加密保存。
自定义备份	允许用户筛选需要备份的文件（路径/名字/时间）。
备份文件比较	比较指定路径的备份文件与源文件之间的差异。

1.2.3 用户类和特性

本软件产品的使用对象是对数据备份有一定的需求，且掌握基本计算机操作的人员。

1.3 运行环境

1.3.1 基本配置

文件备份软件系统所需的基本配置如下：

- (1) 硬件平台：酷睿 12 代 i5 处理器
- (2) 操作系统：Ubuntu 22.04.2 LTS

1.3.2 其他配置

- (1) 内存：16G
- (2) 硬盘：512G
- (3) 网络：无需联网

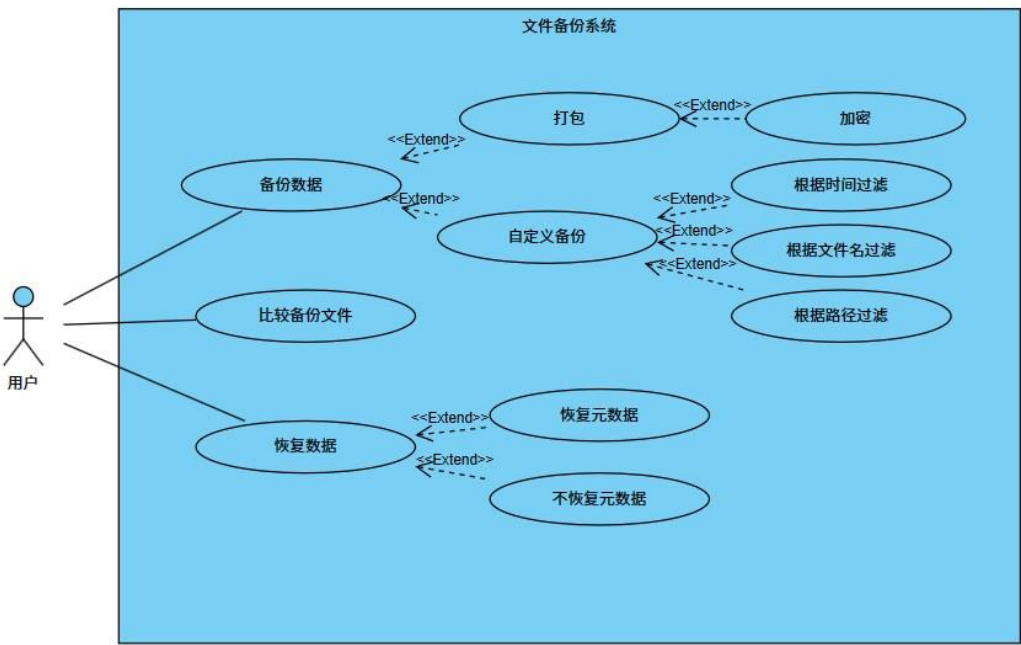
2. 功能需求

2.1 功能划分

本软件为用户提供数据备份、数据还原、打包解包、加密解密、自定义备份和备份文件比较功能。

2.2 系统用例

2.2.1 用例图



2.2.2 用例描述

用例名称	打包文件
范围	数据备份软件
主执行者	用户
涉众及其关注点	用户：希望能准确打包文件
前置条件	无
成功保证	待打包文件路径正确，备份目标路径正确
主成功场景	1. 用户打包文件 2. 文件打包成功，程序正常结束
扩展	1a. 待打包文件路径不正确 1. 提示文件路径不正确并结束程序 1b. 备份目标路径不正确 1. 提示文件路径不正确并结束程序
特殊需求	暂无
其它	暂无

用例名称	加密文件
范围	数据备份软件
主执行者	用户
涉众及其关注点	用户：希望能准确加密文件
前置条件	打包文件
成功保证	待加密文件路径正确，备份目标路径正确
主成功场景	1. 用户加密文件 2. 文件加密成功，程序正常结束
扩展	1a. 待加密文件路径不正确 1. 提示文件路径不正确并结束程序 1b. 备份目标路径不正确 1. 提示文件路径不正确并结束程序

特殊需求	暂无
其它	暂无

用例名称	自定义备份
范围	数据备份软件
主执行者	用户
涉众及其关注点	用户：希望能准确筛选待备份文件
前置条件	打包文件
成功保证	输入的过滤表达式正确，备份目标路径正确
主成功场景	1. 用户自定义备份文件 2. 文件自定义备份成功，程序正常结束
扩展	1a. 输入的过滤表达式不正确 1. 提示过滤表达式不正确并结束程序 1b. 备份目标路径不正确 1. 提示文件路径不正确并结束程序
特殊需求	暂无
其它	暂无

用例名称	比较备份文件
范围	数据备份软件
主执行者	用户
涉众及其关注点	用户：希望能准确得到指定备份文件与源文件的差异
前置条件	文件已备份
成功保证	输入的已备份文件路径正确
主成功场景	1. 用户发起比较请求 2. 程序输出比较结果
扩展	1a. 输入的已备份文件路径不正确 1. 提示路径不正确并结束程序

特殊需求	暂无
其它	暂无
用例名称	恢复文件元数据
范围	数据备份软件
主执行者	用户
涉众及其关注点	用户：希望能恢复文件并恢复文件元数据
前置条件	文件已备份
成功保证	待恢复文件路径正确，恢复目标路径正确
主成功场景	1. 用户请求恢复文件元数据 2. 恢复成功，程序正常结束
扩展	1a. 待恢复文件路径不正确 1. 提示文件路径不正确并结束程序 1b. 恢复目标路径不正确 1. 提示文件路径不正确并结束程序
特殊需求	暂无
其它	暂无

3. 外部接口需求

3.1 用户界面

- (1) 初始化时，每一个步骤都会给予用户简单、易理解的文字提示信息。
- (2) 用户输入不符合规范时，客户端会继续运行并告知用户错误信息，但不会进入下一界面。
- (3) 进行数据备份时，开始数据备份、结束数据备份都会通过客户端窗口显示提示信息告知用户。
- (4) 进行数据还原时，数据还原成功、文件数据损坏都会通过客户端窗口显示提示信息告知用户。
- (5) 其他错误情况时，客户端会继续运行并告知用户错误信息。
- (6) 客户端提示信息的样式为命令行窗口的文字样式。

3.2 硬件接口

无

3.3 软件接口

操作系统：Ubuntu 22.04.2 LTS
函数库：OpenSSL 1.1.1n

4. 其它非功能性需求

4.1 性能需求

- (1) 系统支持的并发操作数量：10
- (2) 响应时间：用户和运维人员进行操作时响应时间在 1 秒以内，备份/还原反馈结果在 3 秒以内。

4.2 安全性需求

因为本软件不涉及服务器，所以无安全性需求。

4.3 软件质量属性

- (1) 易恢复性：在故障发生后，能够重建性能水平，并恢复直接受影响的数据。
- (2) 容错性：在系统出错时，不影响用户的行为操作与数据。
- (3) 成熟性：系统的故障率需要保持在一定的水平下。
- (4) 易学性：系统操作简单，方便用户学习。
- (5) 易用性：系统操作简单，方便用户使用。
- (6) 稳定性：系统应保证正常运行，正确提示相关内容。
- (7) 可移植性：系统可以移植到其他平台使用。

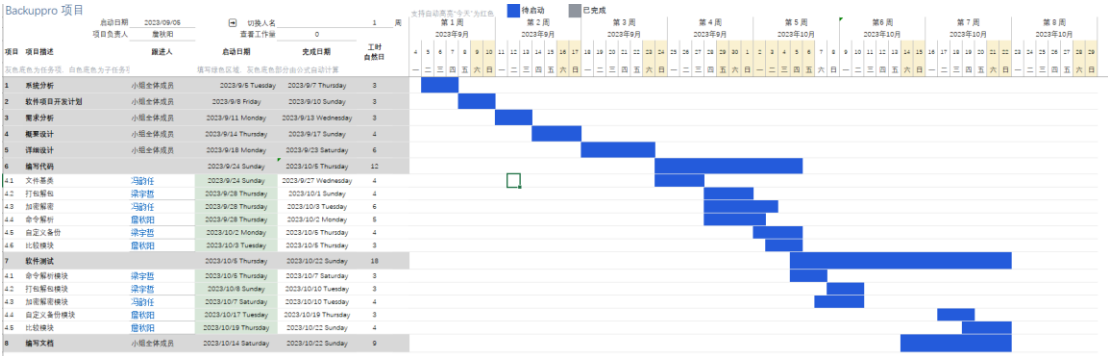
5. 项目规划

5.1 人员分工

项目由三位成员分工具体安排如下。

角 色	主要职责	负责模块	人员	备注
项目经理 PM	<ul style="list-style-type: none">项目全面负责项目设计主要框架/模块编写项目进度控制	<ul style="list-style-type: none">代码框架代码汇总比较模块	詹秋阳	
程序员 DEV	<ul style="list-style-type: none">模块编写	<ul style="list-style-type: none">打包\解包模块过滤模块	梁宇哲	
程序员 DEV	<ul style="list-style-type: none">模块编写	<ul style="list-style-type: none">加密\解密模块文件基类模块	冯韵任	

5.2 项目总体规划



上图是使用甘特图完成的项目总体规划，结合软件工程的规范，项目可以拆分成 8 个任务。前期任务如系统分析、开发计划以及需求分析在九月中旬前完成，随后进入概要设计与详细设计阶段，这两个计划

在一周内完成，最后留有一个月左右的时间进行代码编写及测试。

系统设计文档（20 分）

1. 开发环境和工具

1.1 开发环境

- （1）硬件平台：酷睿 12 代 i5 处理器
- （2）操作系统：Ubuntu 22.04.2 LTS
- （3）编译环境：g++ 11.4.0、cmake 3.22.1
- （4）IDE：Visual Studio Code

1.2 依赖库

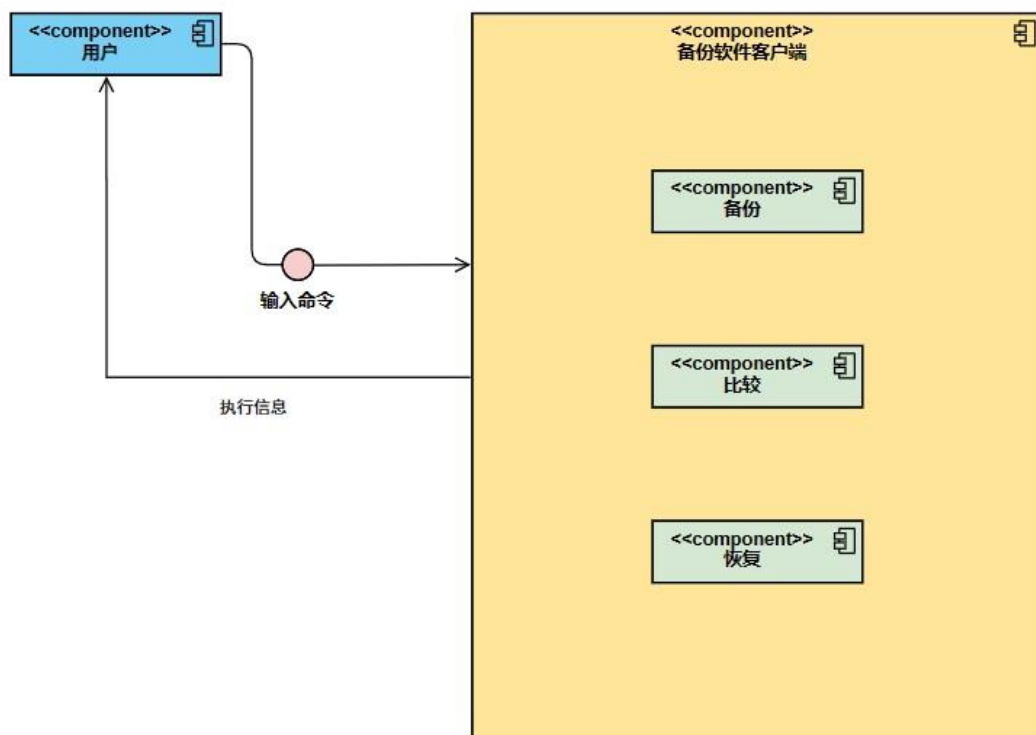
- （1）OpenSSL：版本 1.1.1n，用于加密\解密模块和比较模块开发。
- （2）cxxopts：用于命令行解析参数

2. 总体设计

2.1 系统结构设计

2.1.1 顶层系统结构

2.1.1.1 构件图

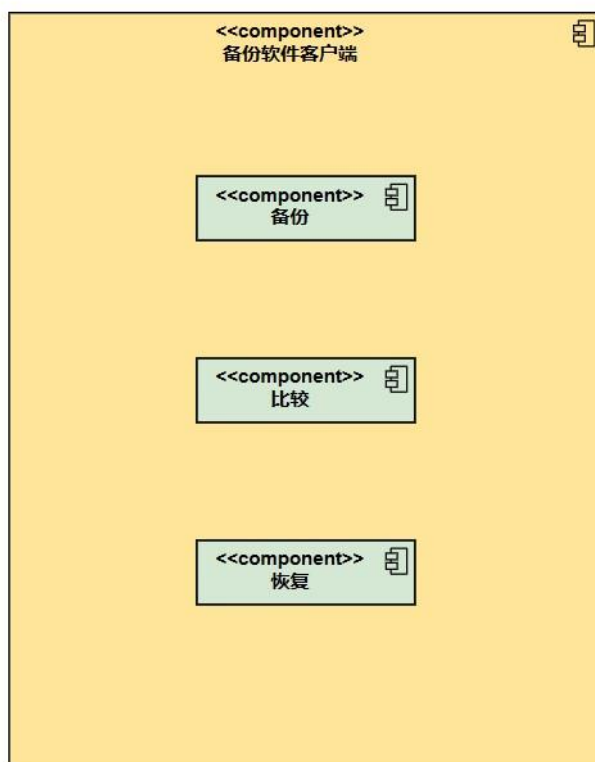


2.1.1.2 构件图描述

用户通过命令行键入命令，发送给备份软件客户端，客户端处理完成后将执行信息返回给用户。

2.1.2 子系统客户端结构

2.1.2.1 构件图

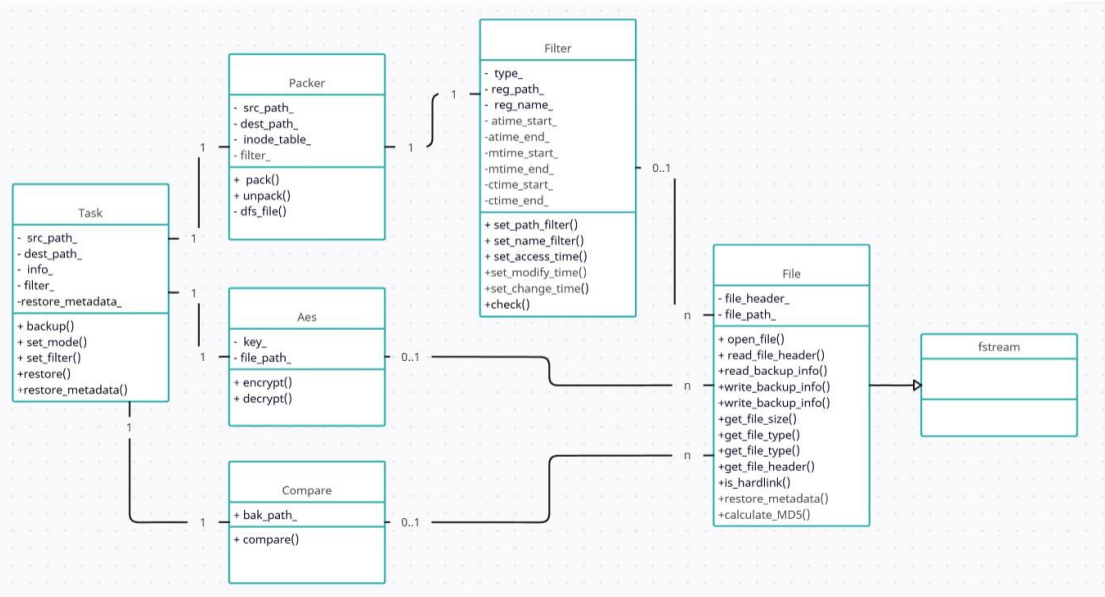


2.1.2.2 构件图描述

备份软件客户端由三个模块组成：备份模块、比较模块和恢复模块。当客户端接收到用户请求时，执行对应的功能模块，并将执行信息输出。

3. 静态建模

3.1 系统对象模型



3.2 类（对象）描述

3.2.1 Aes 类（对象）

3.2.1.1 类描述

该类用于备份文件的加密与解密。

3.2.1.2 属性描述

名字	类型
key_	unsigned char
file_path_	filesystem::path

3.2.1.3 方法描述

方法名	返回类型	返回值	参数	用途
encrypt()	bool	True&False	无	备份文件加密
decrypt()	int	0&-1&-2	无	备份文件解密

3.2.2 CmdParser 类（对象）

3.2.2.1 类描述

该类用于命令行参数解析。

3.2.2.2 属性描述

名字	类型
options_	cxxopts::Options
parse_result_	cxxopts::ParseResult

flag_backup_	bool
flag_restore_	bool
flag_encrypt_	bool
flag_metadata_	bool
flag_help_	bool
flag_compare_	bool
flag_input_	bool
flag_output_	bool
flag_password_	bool
flag_path_	bool
flag_name_	bool
flag_atime_	bool
flag_mtime_	bool
flag_ctime_	bool
arg_compare_	string
arg_input_	string
arg_output_	string
arg_password_	string
arg_path_	string
arg_name_	string
atime_start_	time_t
atime_end_	time_t
mtime_start_	time_t
mtime_end_	time_t
ctime_start_	time_t
ctime_end_	time_t
arg_atime_	string
arg_mtime_	string
arg_ctime_	string

3. 2. 2. 3 方法描述

方法名	返回类型	返回值	参数	用途
cmd_parse	bool	True&False	int argc, char **argv	命令解析
check_password	bool	True&False	无	检查密码格式
print_error_info	void	无	无	输出错误信息
arg_check	bool	True&False	无	检查参数
check_args_backup	bool	True&False	无	检查备份参数
check_args_restore	bool	True&False	无	检查恢复参数
check_args_compare	bool	True&False	无	检查比较参数
convert_time	bool	True&False	std::string time, time_t &time_start, time_t &time_end	转换时间

string_to_time	time_t	t_	std::string time	字符串转时间
check_reg_exp	bool	True&False	无	检查正则表达式

3.2.3 Compare 类（对象）

3.2.3.1 类描述

该类用于对比指定备份文件与源文件之间的差异

3.2.3.3 属性描述

名字	类型
bak_path_	string

3.2.3.2 方法描述

方法名	返回类型	返回值	参数	用途
compare	bool	True&False	无	比较备份文件差异

3.2.4 File 类（对象）

3.2.4.1 类描述

该类作为对 fstream 类的继承，提供了更多接口，方便本软件开发使用。

3.2.4.2 属性描述

名字	类型
file_header_	fileHeader
file_path_	filesystem::path

3.2.4.3 方法描述

方法名	返回类型	返回值	参数	用途
open_file	bool	True&False	std::ios_base::openmode mode	以特定模式打开文件
read_file_header	FileHeader	file_header_	无	读文件头
read_backup_info	BackupInfo	info	无	读备份信息
write_backup_info	void	无	无	写备份信息
write_backup_info	void	无	BackupInfo info	写备份信息
get_file_size	size_t	file_header_.metadata_.st_size	无	读文件尺寸
get_file_type	FileType	File::get_file_type(file_header_)	无	读文件类型

get_file_type	FileType	FILE_TYPE_NORMAL& FILE_TYPE_DIRECTORY& FILE_TYPE_SYMBOLIC_LINK& FILE_TYPE_FIFO& FILE_TYPE_OTHERS	const FileHeader file_header	读文件 类型
get_file_header	FileHeader	file_header_		读文件 头
is_hardlink	bool	True&False	无	判 断 是 否 为 硬 链接
restore_metadata	void	无	无	恢 复 元 数据
calculate_MD5	void	无	std::filesystem::path file_path, unsigned char *md5	计 算 文 件 md5 哈希值

3.2.5 Filter 类（对象）

3.2.5.1 类描述

该类用于实现对文件进行筛选，实现自定义备份功能

3.2.5.2 属性描述

名字	类型
type_	unsigned char
reg_path_	string
reg_name_	string
atime_start_	time_t
atime_end_	time_t
mtime_start_	time_t
mtime_end_	time_t
ctime_start_	time_t
ctime_end_	time_t

3.2.5.3 方法描述

方法名	返回类型	返回值	参数	用途
set_path_filter	void	无	std::string reg_path	设置路径过 滤
set_name_filter	void	无	std::string reg_name	设置文件名 过滤
set_access_time	void	无	time_t atime_start, time_t atime_end	设置访问时 间过滤
set_modify_time	void	无	time_t mtime_start, time_t mtime_end	设置修改时 间过滤

set_change_time	void	无	time_t ctime_start, time_t ctime_end	设置改变时间过滤
check	bool	True&False	const FileHeader &file_header	检查是否满足过滤条件

3.2.6 Packer 类（对象）

3.2.6.1 类描述

该类用于打包解包

3.2.6.2 属性描述

名字	类型
src_path_	filesystem::path
dest_path_	filesystem::path
inode_table_	unordered_map<ino_t, std::string>
filter_	Filter

3.2.6.3 方法描述

方法名	返回类型	返回值	参数	用途
pack	bool	True&False	无	打包
unpack	bool	True&False	bool restore_metadata	解包

3.2.6 Task 类（对象）

3.2.6.1 类描述

该类作为任务类，主要用于调用下层的 Packer、Aes、Compare 等接口，完成用户任务

3.2.6.2 属性描述

名字	类型
src_path_	filesystem::path
dest_path_	filesystem::path
info_	BackupInfo
filter_	Filter
restore_metadata_	bool

3.2.6.3 方法描述

方法名	返回类型	返回值	参数	用途
backup	bool	True&False	std::string password	备份
set_mode	void	无	unsigned char mode	设置备份模式
set_filter	void	无	const Filter filter	设置过滤器
restore	bool	True&False	std::string password	恢复
restore_metadata	void	无	bool	恢复元数据

			restore_metadata	
get_backup_info	bool	True&False	无	得到备份信息
get_backup_mode	unsigned char	info_.mode_	无	得到备份模式
get_backup_info	bool	True&False	std::string file_path, BackupInfo &info	得到备份信息
compare	bool	True&False	无	比较

4. 动态建模

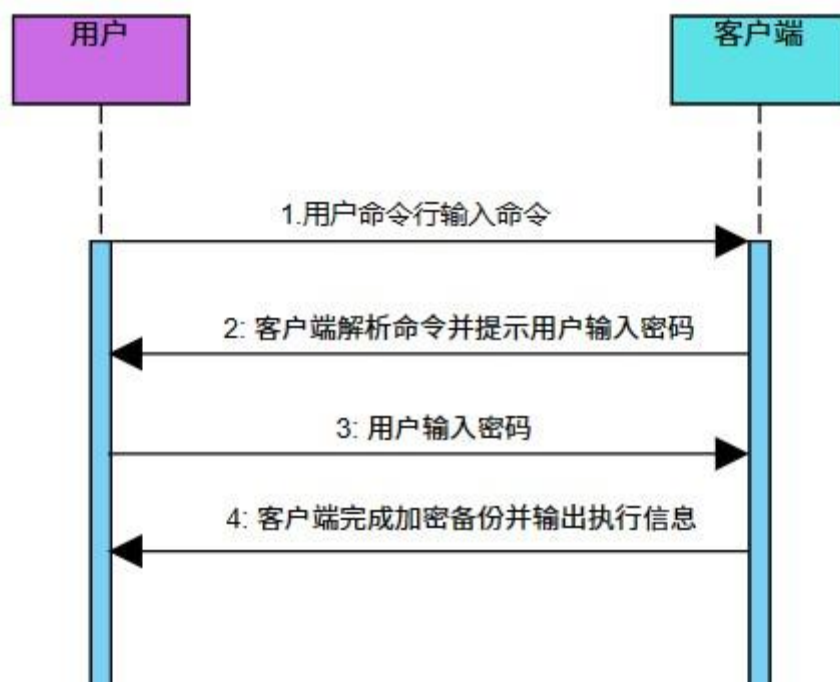
4.1 系统场景

4.1.1 场景：根据路径自定义打包并加密文件

4.1.1.1 场景描述

用户通过命令行命令请求根据路径自定义打包并加密文件，并给出相应过滤规则。

4.1.1.2 顺序图



4.1.1.3 流程说明

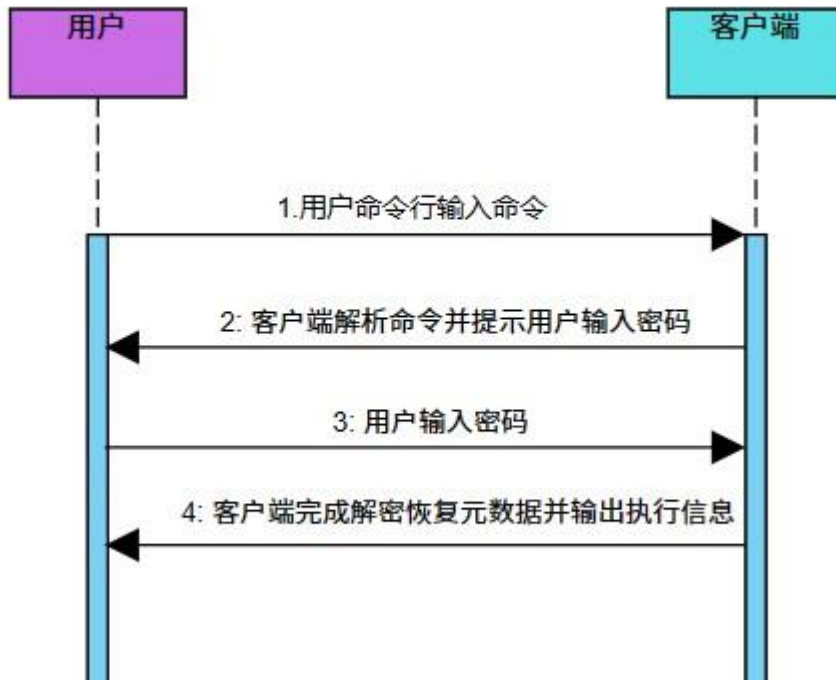
1. 用户命令行输入命令
2. 客户端解析命令并提示用户输入密码
3. 用户输入密码
4. 客户端完成加密备份并输出执行信息

4.1.2 场景：解密文件并恢复文件元数据

4.1.2.1 场景描述

用户通过命令行命令请求解密文件并恢复文件元数据，并给出解密密码。

4.1.2.2 顺序图



4.1.2.3 流程说明

1. 用户命令行输入命令
2. 客户端解析命令并提示用户输入密码
3. 用户输入密码
4. 客户端完成解密恢复元数据并输出执行信息

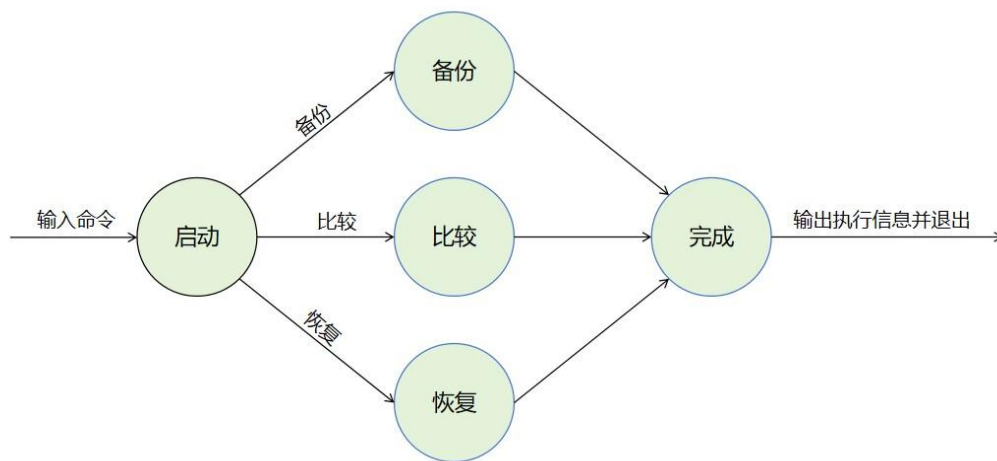
4.2 系统状态

4.2.1 客户端状态切换

4.2.1.1 状态描述

客户端的状态切换

4.2.1.2 状态转换图



4.2.1.3 状态图说明

1. 当接收到外部命令时，客户端进入启动态
2. 如果命令解析为备份，客户端进入备份态
3. 如果命令解析为比较，客户端进入比较态
4. 如果命令解析为恢复，客户端进入恢复态
5. 执行完成后进入完成态
6. 输出执行信息后退出

软件测试报告（20 分）

1. 引言

为了尽可能的找出软件的不足，提高软件的质量，促进软件的成功验收，专门制定了本大纲。其主要目的在于为所要进行的测试工作制定各种必要的准则和规范，以及在有关方面协议的基础上对测试工作进行合理组织与管理。

2. 功能测试

2.1 命令解析模块

本模块测试用例如下：

程序版本号：1.1.1	模块名：命令解析模块
测试用例编号：TestCase-CmdParser-01	用例级别：重要
用例名称：备份选项解析	测试时间：2023/10/5
预置条件：无	
测试输入：源路径、目标路径和需要过滤的文件名正则表达式，以及对应命令行选项	

操作步骤	1) 直接在命令行键入 backuppro -be -i test -o bakdir/test -- name “.cpp\$” 2) 两次输入加密密码，测试时以 2020070902021 为例
预期结果：命令行参数解析成功	
实际输出：命令行参数解析成功	
测试人员：梁宇哲	

程序版本号：1.1.1	模块名：命令解析模块
测试用例编号：TestCase-CmdParser-02	用例级别：重要
用例名称：恢复选项解析	测试时间：2023/10/5
预置条件：无	
测试输入：源路径和目标路径，以及对应命令行选项	
操作步骤	1) 直接在命令行键入 backuppro -ra -i bakdir/test.pak -o resdir/
预期结果：命令行参数解析成功	
实际输出：命令行参数解析成功	
测试人员：梁宇哲	

程序版本号：1.1.1	模块名：命令解析模块
测试用例编号：TestCase-CmdParser-03	用例级别：重要
用例名称：比较备份文件选项解析	测试时间：2023/10/5

预置条件：无	
测试输入：源路径和目标路径，以及对应命令行选项	
操作步骤	1) 直接在命令行键入 backuppro -c bakdir/test.pak
预期结果：命令行参数解析成功	
实际输出：命令行参数解析成功	
测试人员：梁宇哲	

2.2 打包解包模块

本模块测试用例如下：

程序版本号：1.1.1	模块名：打包解包模块
测试用例编号：TestCase-Packer-01	用例级别：重要
用例名称：打包	测试时间：2023/10/5
预置条件：输入的路径名正确	
测试输入：源路径和目标路径，以及对应命令行选项	
操作步骤	1) 直接在命令行键入 backuppro -b -i test -o bakdir/test
预期结果：文件打包成功	
实际输出：文件打包成功	
测试人员：梁宇哲	

程序版本号：1.1.1		模块名：打包解包模块
测试用例编号：TestCase-Packer-02		用例级别：重要
用例名称：解包（不恢复文件元数据）		测试时间：2023/10/5
预置条件：输入的路径名正确		
测试输入：源路径和目标路径，以及对应命令行选项		
操作步骤	1) 直接在命令行键入 backuppro -r -i bakdir/test.pak -o resdir/	
预期结果：文件解包成功，文件元数据与源文件不同		
实际输出：文件解包成功，文件元数据与源文件不同		
测试人员：梁宇哲		

程序版本号：1.1.1		模块名：打包解包模块
测试用例编号：TestCase-Packer-03		用例级别：重要
用例名称：解包（恢复文件元数据）		测试时间：2023/10/5
预置条件：输入的路径名正确		
测试输入：源路径和目标路径，以及对应命令行选项		
操作步骤	1) 直接在命令行键入 backuppro -ra -i bakdir/test.pak -o resdir/	
预期结果：文件解包成功，文件元数据与源文件相同		
实际输出：文件解包成功，文件元数据与源文件相同		

测试人员：梁宇哲

2.3 加密解密模块

本模块测试用例如下：

程序版本号：1.1.2	模块名：加密解密模块
测试用例编号：TestCase-Aes-01	用例级别：重要
用例名称：加密	测试时间：2023/10/7
预置条件：输入的路径名正确	
测试输入：源路径和目标路径，以及对应命令行选项	
操作步骤	1) 直接在命令行键入 backuppro -be -i test -o bakdir/test 2) 两次键入密码，测试密码 123456789
预期结果：文件加密成功	
实际输出：文件加密成功	
测试人员：冯韵任	

程序版本号：1.1.2	模块名：加密解密模块
测试用例编号：TestCase-Aes-02	用例级别：重要
用例名称：解密（不恢复文件元数据）	测试时间：2023/10/7
预置条件：输入的路径名正确	
测试输入：源路径、目标路径和对应的加密密码，以及对应命令行选项	

操作步骤	1) 直接在命令行键入 backuppro -r -i bakdir/test.pak.ept -o resdir/ 2) 键入测试密码 123456789
预期结果: 文件解密成功, 文件元数据与源文件不同	
实际输出: 文件解密成功, 文件元数据与源文件不同	
测试人员: 冯韵任	

程序版本号: 1.1.2	模块名: 加密解密模块
测试用例编号: TestCase-Aes-03	用例级别: 重要
用例名称: 解包 (恢复文件元数据)	测试时间: 2023/10/7
预置条件: 输入的路径名正确	
测试输入: 源路径、目标路径和对应的加密密码, 以及对应命令行选项	
操作步骤	1) 直接在命令行键入 backuppro -ra -i bakdir/test.pak.ept -o resdir/ 2) 键入测试密码 123456789
预期结果: 文件解密成功, 文件元数据与源文件相同	
实际输出: 文件解密成功, 文件元数据与源文件相同	
测试人员: 冯韵任	

2.4 自定义备份模块

本模块测试用例如下:

程序版本号: 1.1.3	模块名: 自定义备份模块
测试用例编号: TestCase-Filter-01	用例级别: 重要

用例名称：根据文件名过滤		测试时间：2023/10/17
前置条件：输入的路径名和正则表达式正确		
测试输入：源路径和目标路径，以及对应命令行选项		
操作步骤	1) 直接在命令行键入 backuppro -b -i test -o bakdir/test --name “\. cpp\$”	
预期结果：只有以.cpp 结尾的文件被备份		
实际输出：只有以.cpp 结尾的文件被备份		
测试人员：詹秋阳		

程序版本号：1.1.3		模块名：自定义备份模块
测试用例编号：TestCase-Filter-02		用例级别：重要
用例名称：根据路径过滤		测试时间：2023/10/17
前置条件：输入的路径名和正则表达式正确		
测试输入：源路径和目标路径，以及对应命令行选项		
操作步骤	1) 直接在命令行键入 backuppro -b -i test -o bakdir/test --path “test/dir/”	
预期结果：只有以在 test/dir/路径下的文件被备份		
实际输出：只有以在 test/dir/路径下的文件被备份		
测试人员：詹秋阳		

程序版本号：1.1.3		模块名：自定义备份模块	
测试用例编号：TestCase-Filter-03		用例级别：重要	
用例名称：根据时间过滤		测试时间：2023/10/17	
预置条件：输入的路径名和时间格式正确			
测试输入：源路径和目标路径，以及对对应命令行选项			
操作步骤		1) 直接在命令行键入 backuppro -b -i test -o bakdir/test --mtime “2023-10-16 13:00:00 2023-10-17 11:00:00”	
预期结果：只有修改时间在 2023-10-16 13:00:00 到 2023-10-17 11:00:00 之间的文件被备份			
实际输出：只有修改时间在 2023-10-16 13:00:00 到 2023-10-17 11:00:00 之间的文件被备份			
测试人员：詹秋阳			

2.5 比较模块

本模块测试用例如下：

程序版本号：1.1.4		模块名：比较模块
测试用例编号：TestCase-Compare-01		用例级别：重要
用例名称：比较（无差异）		测试时间：2023/10/19
预置条件：输入的路径名正确		
测试输入：备份文件路径，以及对应命令行选项		
操作步骤	1) 直接在命令行键入 backuppro -c bakdir/test.pak	

预期结果：无差异
实际输出：无差异
测试人员：詹秋阳

程序版本号：1.1.4	模块名：比较模块
测试用例编号：TestCase-Compare-02	用例级别：重要
用例名称：比较（有差异）	测试时间：2023/10/19
预置条件：输入的路径名正确	
测试输入：备份文件路径，以及对对应命令行选项	
操作步骤	1) 直接在命令行键入 backuppro -c bakdir/test.pak
预期结果：a.cpp 文件有差异	
实际输出：a.cpp 文件有差异	
测试人员：詹秋阳	

3. 测试结果分析

测试结果汇总表如下：

测试模块	测试项目	测试结果
命令解析模块	备份选项解析	测试通过
	恢复选项解析	测试通过
	比较备份文件选项解析	测试通过
打包解包模块	打包	测试通过
	解包（不恢复文件元数据）	测试通过
	解包（恢复文件元数据）	测试通过
加密解密模块	加密	测试通过
	解密（不恢复文件元数据）	测试通过

		解密（恢复文件元数据）	测试通过
	自定义备份模块	根据文件名过滤	测试通过
		根据路径过滤	测试通过
		根据时间过滤	测试通过
	比较模块	比较（无差异）	测试通过
		比较（有差异）	测试通过

七、实验结论：

总的来说，本小组的项目包含了以下功能：

（1）备份与恢复

本功能贯穿设计始终，用户可选择仅打包、打包且加密方式进行备份，恢复时会根据之前保存备份模式进行恢复。因此，本部分的实现方式在下面“（2）打包解包”与“（3）加密解密”中叙述。

（2）打包解包

打包解包功能借鉴了老师课上介绍的 Tar 打包原理。

在本项目中，实现了一个文件基类，将文件拆分成了文件头和文件路径，以便顺利完成打包解包功能。在打包时，程序会遍历整个备份文件，将每个文件的文件头和文件内容写入打包文件。在解包时，会根据文件头中的 size 大小逐个解包文件。

（3）加密解密

加密解密功能的实现借助了 Aes 加密方法。

在本项目中，首先将用户密码加密放在加密文件中，随后对文件内容进行编码，完成加密功能。在解密时，先将用户输入的密码编码后进行比对，然后再依次解码文件，实现解密功能。

（4）自定义备份

自定义备份主要基于正则表达式实现。

本项目中，对于过滤路径、文件名、时间，用户给出对应的正则表达式，然后程序会根据正则表达式进行匹配，实现自定义备份功能。

（5）比较功能

比较功能基于 md5 技术实现。

在之前定义的文件头中加入 md5 成员变量，依照解包的逻辑，依次对比打包文件中每个文件写入的 md5 哈希值与现有文件是否一致即可。在对比过程中，每对比一个文件输出比较结果，从而实现差异文件可视化。

最终项目实现效果良好，在多轮测试中未出现明显 bug，并符合用户友好性原则。

八、总结及心得体会：

回顾整个开发过程，前期效率极其低下是最大的问题。主要原因是前期技术调研不够充分，小组成员还不够熟悉导致交流不深入。

在后期及时调整了开发方案，增加了沟通频率，使得开发效率逐渐提升，最后如期完成任务。

对于我最大的启发，是作为组长不应该太多参与细枝末节的工作，这样会缺乏全局视角，很容易导致小组走弯路。也感谢老师在课上分享的各种知识，给了我很多启发，让我明白了软件开发并不是一件简单的事情，而是团队协作的结晶。

九、对本实验过程及方法、手段的改进建议：

建议老师可以在第一节课增加一些对 linux 基础操作的介绍，因为有部分同学到现在接触 linux 的机会也比较少，更不用谈开发经验，所以希望老师可以适当引入一下。

报告评分：

指导教师签字：