**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** | **张耕（2017120101013）、韦宇霏（2017060106022）、成昊（2017060106002）** |
| 1. **实验室名称：**基础实验大楼A502 | |
| **二、实验项目名称：** 文件备份软件 | |
| **三、实验目的：**  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。 | |
| **四、实验内容：**  制作一款文件备份软件。至少包含以下基本功能：   * 数据的备份（原始数据的打包、压缩） * 数据的恢复（备份数据的解压、解包） * 基本的用户交互界面 * 对已备份完成的文件进行校验   除此之外，该文件备份软件还具备以下自选功能：   * 云盘备份：将数据通过网络上传至服务器，变成一个云盘备份。 * 加密备份：允许用户在备份数据时，为备份数据设置密码，解压时，需要输入密码。 * 自动备份：允许用户设置自动备份，比如每天晚上8：00对指定文件或文件夹进行备份，备份到指定的目录下。 | |
| **五、实验器材（设备、元器件）：**  个人电脑一台 | |
| **六、实验步骤及操作：**  **需求分析：**  **1. 引言**  **1.1 编写目的**  编写此文档的目的是进一步定制软件开发的细节问题，希望能使本软件开发工作更具体。为了使用户、软件开发者及分析和测试人员对该软件的初始规定有一个共同的理解，它说明了本软件的各项功能需求、性能需求和数据需求，明确标识各项功能的具体含义，阐述实用背景及范围，提供客户解决问题或达到目标所需要的条件或权能，提供一个度量和遵循的基准。具体而言，编写软件需求说明的目的是为所开发的软件提出：  a) 软件设计总体要求，作为软件开发人员、软件测试人员相互了解的基础。  b) 功能要求，数据结构和采集要求，重要的接口要求，作为软件设计人员进行概要设计的依据。  c) 软件确认测试的依据。  **1.2 名词定义**  平面文件：一个文件夹中含有一个或多个文件，不含有子目录。  目录树：一个文件夹中含有文件和目录，并且子目录下可能还含有孙目录和文件依次下推。  打包：将平面文件或目录树合并成一个总的文件。  解包：将合并后的文件还原成原来的平面文件或目录树。  压缩：将打包后的文件通过哈夫曼编码算法压缩为压缩文件。  解压：将压缩文件通过哈夫曼编码算法解压为打包文件。  加密：使用密码将压缩文件加密为加密文件。  解密：使用密码将加密文件解密为压缩文件。  上传：将备份文件上传至云服务器。  下载：将云服务器上的备份文件下载到本地。  定时：每天或每周对一些文件进行备份，可以选择加密和上传到云服务器。  校验：对于定时的备份，可以将当前备份与磁盘上的原始文件进行对比，查看差异。  **2 ．任务概述**  **2.1 目标**  本系统旨在通过打包压缩文件、文件比对和解包解压文件，帮助用户实现文件备份、文件还原和文件验证。  基本技术目标包括：  支持各种类型平面文件和目录树的打包和压缩；  支持文件打包时勾选验证选项进行源文件和备份文件的验证；  支持备份文件的解包解压缩；  支持GUI界面操作；  **2.2 运行环境**  **2.2.1 基本配置**  文件备份软件系统所需的基本配置如下：  （1）操作系统 Win7 及以上 Windows系统；  （2）内存 512M 及以上  （3）硬盘空间 40G 及以上  **2.2.2 其他配置**  （1）开发工具：QT 5.11.2及以上  **3．数据描述**  **3.1 静态数据**  无  **3.2 动态数据**  （1）输入数据：源文件地址和目的地址。  （2）输出数据：操作成功或失败后以弹窗的形式出现的提示信息。  **4. 功能需求**  **4.1 功能划分**  本软件为用户提供备份、还原和校验功能。  **4.2功能描述**  **4.2.1 备份**  用户可以备份一个文件或一个目录下的一些文件或目录到指定目录中。  用户可以设定备份密码，需要输入正确密码才能还原备份。  用户可以设定定时备份，每隔固定时间周期就自动备份一次。  用户可以设定云备份，将备份文件上传到云端。  **4.2.2 恢复**  用户可以把备份文件恢复到指定目录中。  用户可以选择从云端的备份文件恢复。  如果备份文件有密码，必须输入正确密码。  **4.2.3 校验**  有设定定时备份的，可以查看备份文件与磁盘中的文件有何差异。  **5. 运行需求**  **5.1 用户界面**        **6．系统用例**  （用例图只是简单地用图描述了一下系统，但对于每个用例，我们还需要有详细的说明，这样就可以让别人对这个系统有一个更加详细的了解，这时我们就需要写用例描述。  对于用例描述的内容，一般没有硬性规定的格式，但一些必须或者重要的内容还是必须要写进用例描述里面的。用例描述一般包括：简要描述（说明）、前置（前提）条件、基本事件流、其他事件流、异常事件流、后置（事后）条件等等。下面说说各个部分的意思：  简要描述：对用例的角色、目的的简要描述；  前置条件：执行用例之前系统必须要处于的状态，或者要满足的条件；  基本事件流：描述该用例的基本流程，指每个流程都“正常”运作时所发生的事情，没有任何备选流和异常流，而只有最有可能发生的事件流；  其他事件流：表示这个行为或流程是可选的或备选的，并不是总要执行它们；  异常事件流：表示发生了某些非正常的事情所要执行的流程；  后置条件：用例一旦执行后系统所处的状态；）  上面内容需要删掉！！！  用例图：    用例描述：   |  |  | | --- | --- | | **用例名称** | 备份 | | **用例描述** | 备份一个文件或一个目录下的一些文件或目录到指定目录中 | | **执行者** | 用户 | | **前置条件** | 用户给出要备份的文件/目录和备份目标目录 | | **基本事件流** | 把文件打包、压缩，写入目标目录 | | **其他事件流** | 加密备份、云盘备份、自动备份 | | **异常事件流** | 弹框提示出错信息，终止备份 | | **后置条件** | 弹框提示成功，目标目录被写入备份文件 |  |  |  | | --- | --- | | **用例名称** | 恢复 | | **用例描述** | 把备份文件恢复到指定目录中 | | **执行者** | 用户 | | **前置条件** | 用户给出备份文件(bak)和恢复目标目录 | | **基本事件流** | 把备份文件解压缩、解包，写入目标目录 | | **其他事件流** | 云盘恢复 | | **异常事件流** | 弹框提示出错信息，终止恢复 | | **后置条件** | 弹框提示成功，目标目录被写入恢复的文件/目录 |  |  |  | | --- | --- | | **用例名称** | 加密备份 | | **用例描述** | 备份时进行加密 | | **执行者** | 用户 | | **前置条件** | 用户给出密码 | | **基本事件流** | 压缩时同时加密文件主体、并写入密码标志位和密码MD5 | | **其他事件流** | 无 | | **异常事件流** | 无 | | **后置条件** | 文件主体被加密，头部有密码标志为和密码MD5 |  |  |  | | --- | --- | | **用例名称** | 加密备份 | | **用例描述** | 备份时进行加密 | | **执行者** | 用户 | | **前置条件** | 用户给出密码 | | **基本事件流** | 压缩时同时加密文件主体、并写入密码标志位和密码MD5 | | **其他事件流** | 无 | | **异常事件流** | 无 | | **后置条件** | 文件主体被加密，头部有密码标志为和密码MD5 |   **7．项目规划**  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 | * 项目全面负责 * 项目设计 * 主要框架/模块编写 * 项目进度控制 | * 代码框架 * 代码汇总 | 张耕 |  | | 程序员  DEV | * 模块编写 | * 打包\解包模块 * 校验模块 | 成昊 |  | | 程序员  DEV | * 模块编写 | * 压缩\解压 | 韦宇霏 |  |     **软件设计：**  **1.****开发环境和工具**   |  |  | | --- | --- | | 工具 | 作用 | | QT | 界面显示和模块开发 | | Python | 云端文件服务器开发 | | git | 文件汇总和版本控制 | | Star UML | UML绘制 |   **2.概念术语描述**  平面文件：一个文件夹中含有一个或多个文件，不含有子目录。  目录树：一个文件夹中含有文件和目录，并且子目录下可能还含有孙目录和文件依次下推。  打包：将平面文件或目录树合并成一个总的文件。  解包：将合并后的文件还原成原来的平面文件或目录树。  压缩：将打包后的文件通过哈夫曼编码算法压缩为压缩文件。  解压：将压缩文件通过哈夫曼编码算法解压为打包文件。  加密：使用密码将压缩文件加密为加密文件。  解密：使用密码将加密文件解密为压缩文件。  上传：将备份文件上传至云服务器。  下载：将云服务器上的备份文件下载到本地。  定时：每天或每周对一些文件进行备份，可以选择加密和上传到云服务器。  校验：对于定时的备份，可以将当前备份与磁盘上的原始文件进行对比，查看差异。  **3.基本设计描述**  **3.1系统总体逻辑构件图**  G:\软件开发综合实验\ComponentDiagram.png  **3.2主要界面流程描述**  **3.2.1 备份功能流程**  **流程图：**  **G:\软件开发综合实验\备份流程图.png**  **时序图：**  G:\软件开发综合实验\备份.png  **3.2.2 恢复功能流程**  **流程图：**    **时序图：**  G:\软件开发综合实验\恢复.png  **3.2.3 校验功能流程**  **流程图：**    **时序图：**  G:\软件开发综合实验\校验.png  **3.3模块列表**   |  |  |  | | --- | --- | --- | | 模块名称(英文) | 功能 | 备注 | | 备份（backup） | 将所选文件和目录打包为一个文件后再压缩，可以设置密码、定时或上传到云服务器。 | 支持目录树打包，但所选的文件和目录必须位于同一目录下。 | | 恢复（restore） | 将本地或云服务器上的备份文件解压、解包恢复成原来的文件和目录。 | 云服务器上的备份文件会先下载到本地后再恢复。 | | 云盘服务器（cloud server） | 支持查询云端备份文件列表、上传、下载和删除云端文件。 | 该部分使用Python实现。 | | 定时（timer） | 按照一定的频率自动进行备份，可以设置密码或上传到云服务器。 | 定时任务会被添加到“任务”界面。 | | 校验（check） | 对于定时的备份，可以将当前备份与磁盘上的原始文件进行对比，查看差异。 | 该功能在“任务”界面，右键定时任务列表中的表项，在菜单中可以找到。 |   **3.4程序框架**  **G:\软件开发综合实验\ClassDiagram1.png**  **3.4.1 Pack类**  负责将文件和目录树打包为tar文件。  **3.4.2 Unpack类**  负责将tar文件解包为文件和目录树。  **3.4.3 Compressor类**  负责将tar文件使用哈夫曼编码算法压缩为bak文件，可以同时使用密码加密。  **3.4.4 Decompressor类**  负责将bak文件使用哈夫曼编码算法解压为tar文件，如果是加密的bak文件需要提供密码解密。  **3.4.5 TaskManager类**  负责管理定时任务，支持定时任务的添加、删除、修改，同时维护配置文件。  **3.4.6 Widget类**  负责UI绘制和与用户交互等前端逻辑，调用其他类实现备份、恢复、加密、定时、云服务器和校验等功能。  **3.4.7 Check类**  对于定时的备份，可以将当前备份与磁盘上的原始文件进行对比，查看差异。  **3.5数据结构**  **3.5.1 haffNode类**  unsigned long long freq; // 待编码字符出现频率  unsigned char uchar; // 待编码字符  string code; // 编码后的01串  struct haffNode\* left; // 哈夫曼树节点左孩子  struct haffNode\* right; // 哈夫曼树节点右孩子  **3.5.2 Task类**  QList<QString> files; // 需要备份的文件和目录列表  QString backupFilename; // 备份文件的路径  int frequency; // 备份频率  QString password; // 加密密码  bool cloud; // 是否上传到云端  QDateTime nextTime; // 下一次备份时间  **3.6接口规范**  **3.6.1 <Pack类 API>**  **描述**  负责将文件和目录树打包为tar文件。  **3.6.1.1 pack**  **int pack(vector<string>& filepathList, string tarFilename)**  pack函数用于将传入的filepathList中的文件或目录打包成tar文件，写到tarFilename这个文件中。  成功返回0，失败返回相应的错误码。  **3.6.2 <Unpack类 API>**  **描述**  负责将tar文件解包为文件和目录树。  **3.6.2.1 pack**  **int unpack(string tarFilename, string destinationDictory)**  unpack函数用于将传入的**tarFilename**这个tar文件恢复成原来的文件和目录树，写到**destinationDictory**这个文件夹中。  成功返回0，失败返回相应的错误码。  **3.6.3 <Compressor类 API>**  **描述**  负责将tar文件使用哈夫曼编码算法压缩为bak文件，可以同时使用密码加密。  **3.6.3.1 encode**  **void encode(haffNode\* pn, string code)**  递归遍历pn（初始为根节点），为哈夫曼树的叶节点生成哈夫曼编码。  **3.6.3.2 compress**  **int compress(string sourcePath, string destinationPath, string pw = "")**  将sourcePath这个tar文件压缩为destinationPath下与sourcePath同名的bak文件，使用pw加密，如果pw为空则不加密。  返回值：  0：正常执行  1：源文件扩展名不是tar  2：打开源文件失败  3：打开目标文件失败  **3.6.4 <Decompressor类 API>**  **描述**  负责将bak文件使用哈夫曼编码算法解压为tar文件，如果是加密的bak文件需要提供密码解密。  **3.6.4.1 decompress**  **int decompress(string sourcePath, string destinationPath, string pw = "")**  用于将传入的**sourcePath**这个bak文件解压为tar文件，写到**destinationPath**这个文件夹中。使用pw进行解密。  返回值：  0：正常执行  1：源文件扩展名不是bak  2：打开源文件失败  3：打开目标文件失败  4：文件过短，频率表不完整  5：文件结尾不完整  6：密码错误  7：解码错误  **3.6.5 <TaskManager类 API>**  **描述**  负责管理定时任务，支持定时任务的添加、删除、修改，同时维护配置文件。  **3.6.5.1 init**  **void init()**  检测当前目录下是否存在config.json配置文件，若存在则读取其中存储的定时任务配置，初始化taskList列表；若不存在则创建空的config.json文件。  **3.6.5.2 addTask**  **void addTask(Task task)**  将task添加到taskList中。调用writeJson更新配置文件。  **3.6.5.3 removeTask**  **void removeTask(int index)**  将taskList中第index个任务删除。调用writeJson更新配置文件。  **3.6.5.4 clear**  **void clear()**  清空taskList。调用writeJson更新配置文件。  **3.6.5.5 updateTime**  **void updateTime(int index, QDateTime nextTime)**  将taskList中第index个任务的下次执行时间更新为nextTime。调用writeJson更新配置文件。  **3.6.5.6 getTaskList**  **const QList<Task>& getTaskList()**  返回taskList。  **3.6.5.7 writeJson**  **void writeJson()**  将taskList中记录的定时任务格式化为json写入config.json文件中。  **3.6.6 <Widget类 API>**  **描述**  负责UI绘制和与用户交互等前端逻辑，调用其他类实现备份、恢复、加密、定时、云服务器和校验等功能。  **3.6.6.1 Widget**  **explicit Widget(QWidget\* parent = nullptr);**  构造函数，初始化taskManager，填充“任务”界面的定时任务列表。启动定时器，定时执行备份任务。  **3.6.6.2 on\_passwordCheckBox\_stateChanged**  **void on\_passwordCheckBox\_stateChanged(int arg1);**  passwordCheckBox被选中时，passwordLineEdit才可以编辑。  **3.6.6.3 on\_browseButton\_clicked**  **void on\_browseButton\_clicked();**  打开浏览目录对话框，将backupFileDirectoryLineEdit的内容设置为返回的目录。  **3.6.6.4 on\_addFileButton\_clicked**  **void on\_addFileButton\_clicked();**  打开浏览文件对话框，将返回的文件去重后加入backupFileList。  **3.6.6.5 on\_deleteFileButton\_clicked**  **void on\_deleteFileButton\_clicked();**  删除backupFileList当前选择项。  **3.6.6.6 on\_clearFileButton\_clicked**  **void on\_clearFileButton\_clicked();**  清空backupFileList。  **3.6.6.7 on\_addDirectoryButton\_clicked**  **void on\_addDirectoryButton\_clicked();**  打开浏览目录对话框，将返回的目录去重后加入backupFileList。  **3.6.6.8 on\_startBackupButton\_clicked**  **void on\_startBackupButton\_clicked();**  开始备份。检查需要备份的文件、备份文件名、备份保存目录、密码等合法，且需要备份的文件都位于同一目录下，后才开始备份。如果定时被选中，则将该任务加入到定时列表中。调用打包压缩加密，如果上传到云端被选中，则将备份文件上传。  **3.6.6.9 on\_localGroupBox\_clicked**  **void on\_localGroupBox\_clicked(bool checked);**  实现localGroupBox与cloudGroupBox选中互斥。  **3.6.6.10 on\_cloudGroupBox\_clicked**  **void on\_cloudGroupBox\_clicked(bool checked);**  实现localGroupBox与cloudGroupBox选中互斥。拉取云端文件列表，更新显示。  **3.6.6.11 on\_browseLocalFile\_clicked**  **void on\_browseLocalFile\_clicked();**  打开浏览文件对话框，将localFileRestoreLineEdit的内容设置为返回的文件路径。  **3.6.6.12 on\_cloudFileList\_currentItemChanged**  **void on\_cloudFileList\_currentItemChanged(QTreeWidgetItem\* current, QTreeWidgetItem\* previous);**  当cloudFileList当前选择表项改变时，将cloudFileRestoreLineEdit的内容设置为选中的表项（即云端文件名）。  **3.6.6.13 on\_browseRestoreDirectoryButton\_clicked**  **void on\_browseRestoreDirectoryButton\_clicked();**  打开浏览目录对话框，将backupFileRestoreDirectoryLineEdit的内容设置为返回的目录路径。  **3.6.6.14 on\_passwordCheckBox\_2\_stateChanged**  **void on\_passwordCheckBox\_2\_stateChanged(int arg1);**  passwordCheckBox\_2被选中时，passwordLineEdit\_2才可以编辑。  **3.6.6.15 on\_deleteTaskButton\_clicked**  **void on\_deleteTaskButton\_clicked();**  删除taskList中当前选中项，同时也调用taskManager.removeTask将该task删除。  **3.6.6.16 on\_clearTaskButton\_clicked**  **void on\_clearTaskButton\_clicked();**  清空taskList，同时也调用taskManager.clear将taskManager中所有记录的task清空。  **3.6.6.17 on\_taskList\_customContextMenuRequested**  **void on\_taskList\_customContextMenuRequested(const QPoint& pos);**  taskList表项右键单击弹出菜单，有“打开备份文件所在目录”与“与原文件校验”两个选项。分别调用相应函数。  **3.6.6.18 on\_startRestoreButton\_clicked**  **void on\_startRestoreButton\_clicked();**  在检查要恢复的备份文件、要恢复到的目录、密码等合法后才开始恢复。如果是从云端恢复，则需要先从云端把备份文件下载到本地。调用解压解密解包，将恢复的文件写到指定目录。  **3.6.6.19 on\_cloudFileList\_customContextMenuRequested**  **void on\_cloudFileList\_customContextMenuRequested(const QPoint& pos)**  云文件列表表项右键单击弹出菜单，可以删除选择的云备份文件。  **3.6.7 <Check类 API>**  **描述**  对于定时的备份，可以将当前备份与磁盘上的原始文件进行对比，查看差异。  **3.6.7.1 check**  **vector<pair<string, int> > check(vector<string>& files, string directory);**  检测files中的各个文件或目录 是否与directory下的文件有差异，返回差异，返回差异为pair的列表，每个pair的first是文件名；second是差异类型，0为删除，1为修改。若无差异返回空vector。  **软件测试：**  本部分代码见附录。 | |
| **八、实验数据及结果分析：**  **1.运行环境**  系统：Win10  软件：QT 5.11.2  **2.功能测试**  **2.1 xxx模块**  **2.2 xxx模块** | |
| **九、实验结论：**  所编制软件运行结果正确，文档规范。 | |
| **十、总结及心得体会：**  根据学生所负责的模块进行填写。 | |
| **十一、对本实验过程及方法、手段的改进建议：**  无。 | |
| **报告评分：**  **指导教师签字：** | |