

1) 实验中遇到的问题及解决办法说明

在 Linux 环境下安装软件依赖包。

```
[root@ecs-5045 ~]# yum install -y autoconf gcc gcc-c++ libaio-devel flex bison ncurses-devel readline-devel dkms python3-devel pam-devel libffi-devel automake byacc cmake openssl-devel libtool-devel
```

在 Linux 环境下安装 Git 工具并配置 Git 环境。

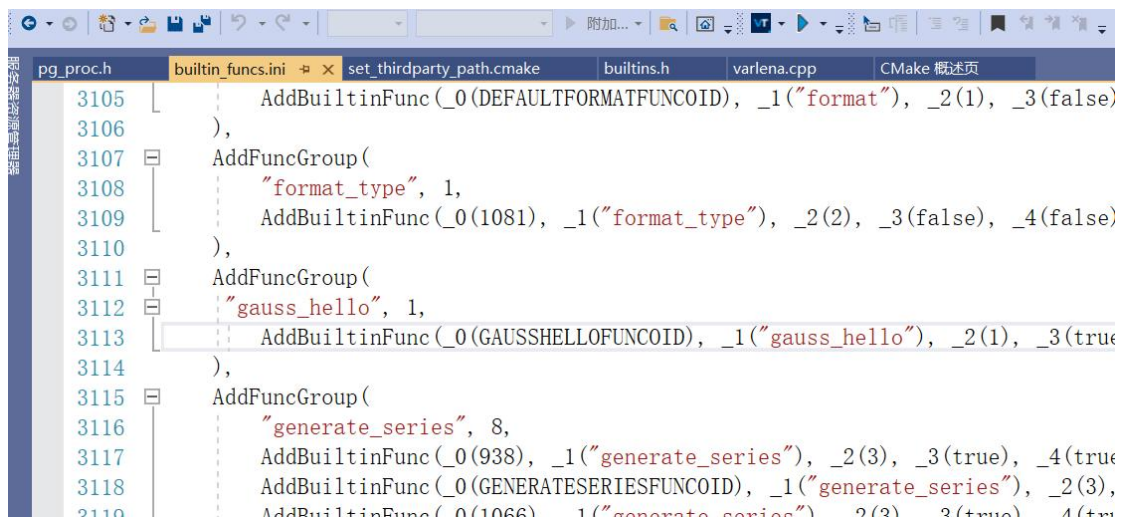
```
[root@ecs-5045 ~]# yum install git -y    ---安装 git 工具
[root@ecs-5045 ~]# git config --global user.name "Your Name"    --- 配置
[root@ecs-5045 ~]# git config --global user.email "email@example.com"    --- 配置
```

下载时报错: *error:GPG check Failed*

修改

```
[root@ecs-5045 ~]# yum install -y autoconf gcc gcc-c++ libaio-devel flex bison ncurses-devel glibc-devel patch readline readline-devel dkms python3-devel pam-devel libffi-devel automake byacc cmake openssl-devel libtool-devel --nogpgcheck
[root@ecs-5045 ~]# yum install git -y --nogpgcheck    ---安装 git 工具
```

(2) 实验耗费时长说明 (各主要步骤大致所需时间)



2.搭建开发环境, 10min

3.二次开发 (gauss_hello 函数为例), 15min,

4. 编译环境要求:约 45min,

5. 验证二次开发结果 20min

(3)openGauss 其他二次开发创意说明(实现的或未实现的二次开发创意说明均可, 实现的创意可说明具体实现步骤, 未实现的创意可说明拟实现思路)。

基于 openGauss 的机器学习平台, 可以基于 openGauss 实现机器学习算法和模型的训练和部署。(只是想法)

数据预处理: 使用 openGauss 提供的数据处理功能, 对原始数据进行清洗、归一化、特征提取等预处理操作。

数据存储: 将预处理后的数据存储到 openGauss 数据库中, 使用 openGauss 的高性能和可扩展性, 为后续的机器学习处理提供支持。

模型训练: 使用 openGauss 提供的数据处理和机器学习算法, 对存储在 openGauss 数据库中的数据进行模型训练。

模型部署: 将训练好的模型部署到 openGauss 数据库中, 以便后续的机器学习处理能够直接使用。

数据分析和预测: 使用 openGauss 提供的机器学习算法和模型, 对存储在 openGauss 数据库中的数据进行分析和预测, 为用户提供数据分析和决策支持等功能。