交付与研发迭代单分支隔离方案

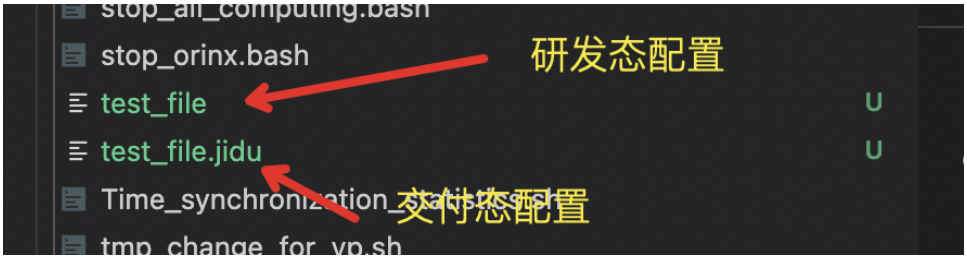
1.背景：

为解决单分支交付与研发迭代功能冲突问题，现在支持通过配置文件隔离交付态和研发态功能，主要包括文件和函数功能的支持

2.方案

- 研发态和交付态配置参数不同

为支持仅参数设置不同问题导致交付和研发存在差异，RD们可以在代码库中准备两份配置文件，研发态配置文件名称不变，交付态配置文件后缀名为.jidu，交付打包时用交付态文件覆盖研发态文件



Bash | 收起

```bash
function updata_jidu_conf() {
   for file in $(find $OUTPUT/cybertron/ -name "*.jidu");do
       jidu_file=$file
       dev_file=${file%.jidu*}
       if [ -f $dev_file ];then
           # 对具有研发态和交付态区别的文件进行覆盖操作
           rm -f $dev_file
           mv $jidu_file $dev_file
       fi
    done
}
```

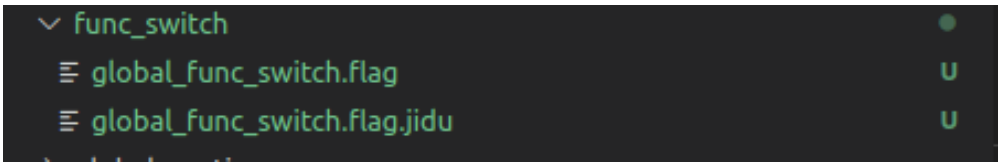
注意：

在复制.jidu文件时， 请保证配置的flag等开关和参数正确适配集度车辆（之前在车端对配置的修改迁移到release.bcloud中了，可参照部分sed指令修改.jidu文件）

- 交付态和研发态具有功能方面差异

开发中的功能不想进入交付态中，RD可以通过gflag进行控制，具体步骤如下：

1.在onboard/conf/func_switch目录下配置了交付和研发的功能开关控制flag文件



各模块可以按照需求设置功能控制的开关参数，区分开发的功能是否进行交付

2.功能开关控制gflag的定义文件在modules/common/configs/global_func_switch.cc和modules/common/configs/global_func_switch.h

使用功能参数前在此文件中定义好要用的参数，例如：

modules/common/configs/global_func_switch.h Bash | 收起

```
#pragma once
#include "gflags/gflags.h"

////////////////////////////////////////////////
//                   switch
//
//       Delivery and development function     //
//              isolation configuration                        //
////////////////////////////////////////////////

DECLARE_bool(check_buf_exceeded);
```

modules/common/func_switch/global_func_switch.cc Bash | 收起

```
#include "modules/common/configs/global_func_switch.h"

DEFINE_bool(check_buf_exceeded, false, "enable_check_buf_exceeded");
```

3.各模块开发功能时引入global_func_switch即可利用定义好的开关参数实现交付态和研发态功能隔离

Bash | 收起

```
1.本模块的BCLOUD引入头文件位置和链接库路径
INCPATHS('$WORK_ROOT/baidu/adu-lab/andes')
Libs("$OUT/so/libanp_configs.so")
2.使用gflags前引入头文件
#include "modules/common/configs/global_func_switch.h"
3.利用定义好的开关参数实现交付态和研发态功能隔离
if (FLAGS_******) {
   # 新功能研发
}
```

4.新增流水线仅用于交付

Bash | 收起

```
- profile:
  name: aarch64_orinx_JIDU_6050os
  mode: BCLOUD
  build:
    command: bcloud build --compiler=gcc930-aarch64-glibc-2.31-ubuntu18-gnu --disable-cc-cxx --exclude-artifact-type='ut' --no-output --profile-name=aarch64_orinx_JIDU_6050os --set-bcloud-macro=USE_CUDA11,UBUNTU20,HDMAP_310LC_FLAG,ANP_MAP,LC_HDMAP_SIMULATOR,DRIVE_OS_6050 --cppflags=-DDRIVE_OS_6050 --enable-global-flags
  artifacts:
    release: true
```

Bash | 收起

```
aarch64_orinx_JIDU_6050os)
        release
        package
        updata_jidu_conf # 隔离交付态和研发态功能
        update_params
        ;;
```