Sommaire

[Sommaire 1](#_Toc36285985)

[1. Connaissances pré-requises 2](#_Toc36285986)

[2. Description 2](#_Toc36285987)

[3. Récupération du dépôt CRDAP sous GitLab 3](#_Toc36285988)

[3.1.1. Sous forme d’archive à installer 3](#_Toc36285989)

[3.1.2. En ligne de commande 5](#_Toc36285990)

[3.1.3. Sous Windows avec Tortoise Git 6](#_Toc36285991)

[4. Traitement des données CTD/LADCP 8](#_Toc36285992)

[4.1. Matlab 8](#_Toc36285993)

[4.1.1. Installation 8](#_Toc36285994)

[4.1.2. Utilisation 9](#_Toc36285995)

[4.2. DOS 11](#_Toc36285996)

[4.2.1. Installation des programmes et outils 11](#_Toc36285997)

[4.2.2. Copie des fichiers bruts Seabird 12](#_Toc36285998)

[4.2.3. Utilisation 12](#_Toc36285999)

[5. Traitements globaux 14](#_Toc36286000)

[5.1. Installation de WSL 14](#_Toc36286001)

[5.2. Configuration 16](#_Toc36286002)

[5.3. Utilisation 18](#_Toc36286003)

[5.3.1. Génération des fichiers ASCII d’entêtes, data et XML 18](#_Toc36286004)

[5.3.2. Génération du fichier NetCDF 19](#_Toc36286005)

[5.4. Automatisation 20](#_Toc36286006)

[5.4.1. Scripts de copie 20](#_Toc36286007)

[5.4.2. Scripts de traitements 21](#_Toc36286008)

[5.4.3. Crontab 23](#_Toc36286009)

[6. Suivi des versions de ce document 23](#_Toc36286010)

# Connaissances pré-requises

L’utilisateur de ce protocole doit maitriser parfaitement l’utilisation et la configuration des systèmes d’exploitations Linux et Windows. Il doit savoir utiliser et configurer les logiciels Seabird Seasave et Sbe-dataprocessing.

# Description

Lorsqu’une campagne océanographique est organisée sur une navire océanographique, l’équipe scientifique déploie son propre matériel et va utiliser et collecter les données des instruments installés sur le navire.

Liste des équipements utilisés lors de campagnes de physique :

* Scientifique :
  + CTD + rosette prélèvements / LADCP
  + Salinomètre
  + Oxygène dissous
  + …..
* Bords :
  + SADCP : ADCP(s) de coque
  + XBT
  + Thermosalinographe
  + Météo
  + Ferry-box
  + GPS
  + Centrale d’attitude
  + Sondeurs acoustiques EK60/80
  + ......

L’acquisition des instruments est généralement réalisée soit par un PC d’acquisition dédié, ce qui est le cas pour la CTD/LADCP, les SADCP, XBT, EK60 ou par une centrale d’acquisition dédiée comme c’est le cas sur les navires opérés par Genavir pour la centrale de navigation CINNA et d’acquisition TECHSAS. Des outils logiciels supplémentaires permettent également la collecte de données agrégées comme c’est le cas avec le cahier de quart informatisé CASINO ou la transmission temps réel des données TSG/Météo avec COLCOR.

Afin de faciliter la collecte et le traitement en temps réel de l’ensemble des données acquises, nous avons mis au point un Système d’Information (SI) déployé sur un espace disque commun ou les données seront automatiquement copiés, traitées puis converties dans des formats génériques et standardisés, que ce soit en ASCII ou en NetCDF OceanSITES.

L’automatisation de ces taches est réalisée sous Linux. Les premières versions étaient réalisées avec Cygwin sous Windows, puis avec l’utilisation de machines virtuelles, et plus récemment, soit avec Docker, soit avec un PC Linux dédié.

Dernièrement nous utilisons directement Linux sous Windows 10, Microsoft fournissant sur son OS, **Windows Subsystem for Linux (WSL)** qui est une couche de compatibilité permettant d'exécuter des exécutables binaires Linux de manière native sur Windows 10.

Ce document détaillera uniquement l’installation et la configuration des outils sous WSL.

Les outils utilisés dans le SI sont principalement écrit en langage Perl car développés il y a maintenant une vingtaine d’année. Le portage est progressivement réalisé en Python et remplacera le code Perl au fur et à mesure des développements. Quelques logiciels compilés sont développés en langage Go (Golang sur Google).

L’ensemble du code et des fichiers de configuration est maintenu avec Git via le dépôt CRDAP pour Cruise Repository for Data Acquisition and Processing.

Le principe de ce projet est qu’à partir d’un dépôt commun contenant la totalité du SI, nous maintenons une branche par navire, branche qui est adaptée et mise à jour en suivant la configuration et la mise à disposition des équipements.

# Récupération du dépôt CRDAP sous GitLab

### Sous forme d’archive à installer

Le code du système d’information (SI) CRDAP pour est disponible sous:

<https://git.outils-is.ird.fr/US191/CRDAP/tree/master>

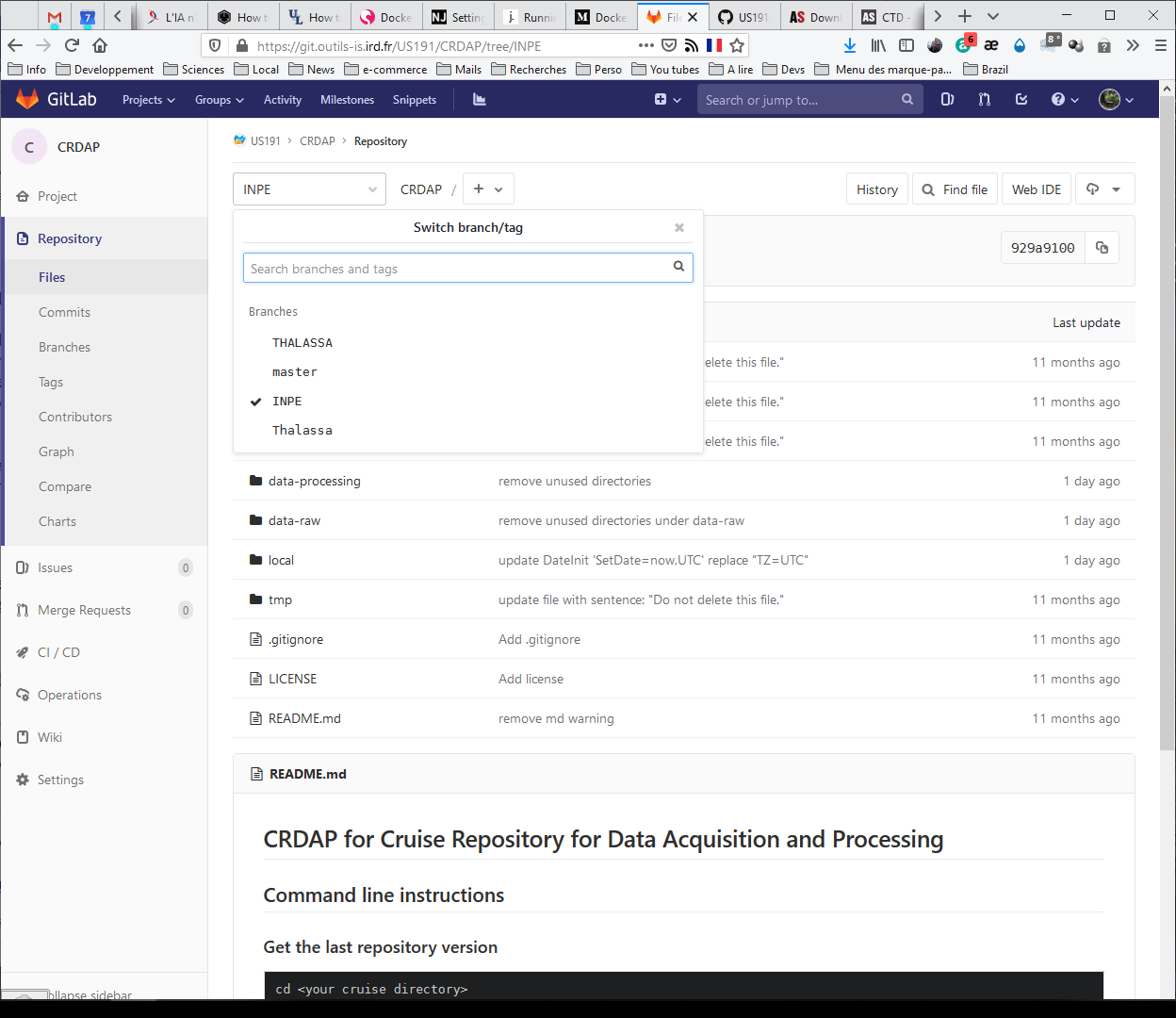


Figure 1CRDAP, sélection de la branche

Puis télécharger l’archive au format désiré :

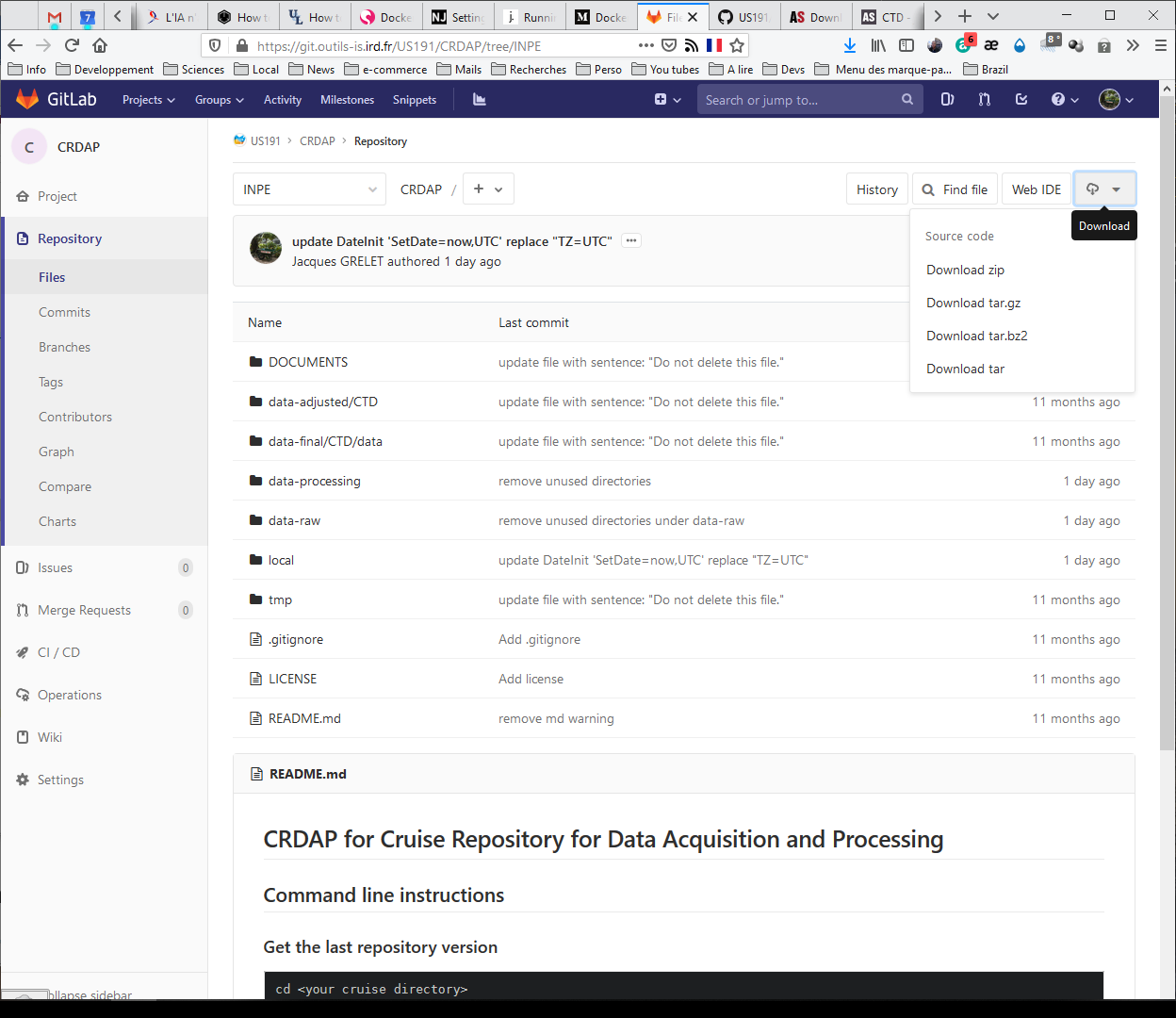


Figure 2: CRDAP, téléchargement de l'archive

### En ligne de commande

Si vous avez déjà Linux ou Git d’installé sur votre PC, vous pouvez cloner directement le dépôt en ligne de commande, et basculer sur la branche désirée. En supposant que votre répertoire de travail est sous c:\PIRATA, soit sous WSL : /mnt/c/PIRATA, nous allons cloner le SI CRDAP avec le nom de la campagne, soit PIRATA-BR18 dans cet exemple.

cd /mnt/c

mkdir PIRATA

cd PIRATA/

git clone https://git.outils-is.ird.fr/US191/CRDAP.git PIRATA-BR18

cd PIRATA-BR18/

git branch

git checkout INPE

git branch

cp local/etc/skel/.bashrc.CRUISE local/etc/skel/.bashrc.PIRATA-BR18

. ~/.bashrc

CTD

[NTIRD-US191-JG4:jgrelet]/mnt/c/PIRATA/PIRATA-BR18/data-processing/CTD

> alias ctd

alias ctd='perl ctd-all.pl --cycle\_mesure=PIRATA-BR18 --institut=INPE --plateforme="VITAL DE OLIVEIRA" --sn=09P-1263 --type=SBE911+ --pi=BRUTO --begin\_date=01/03/2019 --end\_date=05/04/2018 data/asc/PIRA\_\*.hdr --echo --dtd=local --top --all'

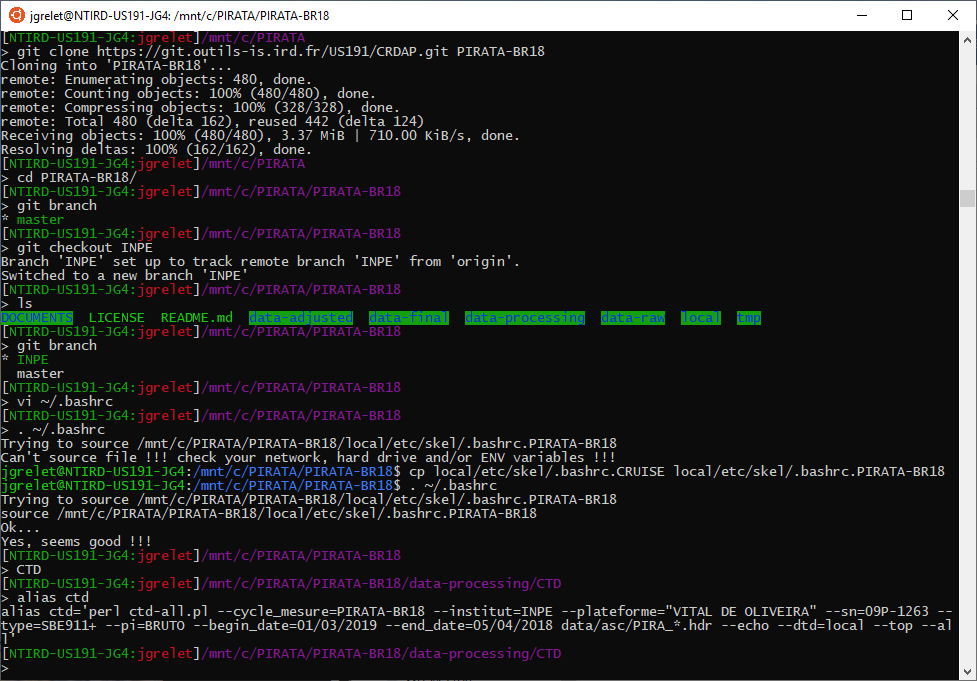


Figure 3:récupération du dépôt en ligne de commande

### Sous Windows avec Tortoise Git

L’utilisateur devra avoir au préalable installé Git et Tortoise Git sous Windows.

On récupère directement le dépôt depuis la branche voulue, INPE dans cet exemple, THALASSA pour les campagnes PIRATA

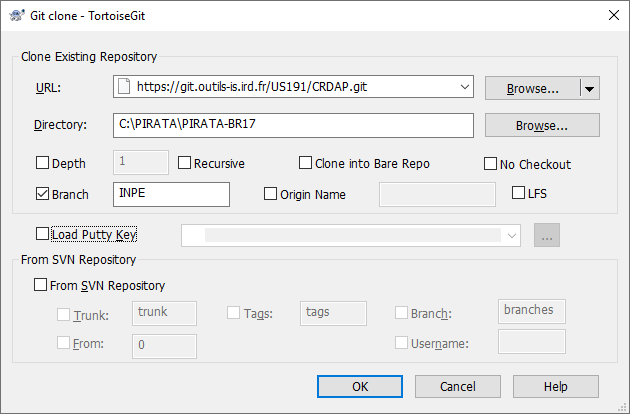
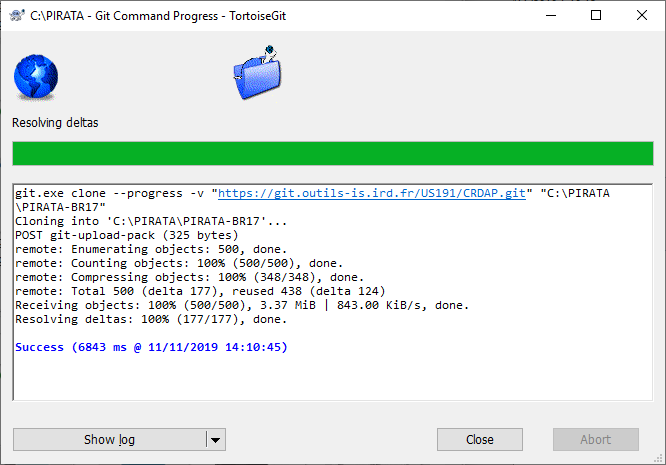


Figure 1: Git clone suous Windows



# Traitement des données CTD/LADCP

Pour l’automatisation de l’archivage et du traitement des données CTD/LADCP, nous avons 2 solutions disponibles. La plus récente fonctionne sous Matlab et lorsqu’elle est installée et configurée, à l’avantage d’être très simple à utiliser en routine pour des utilisateurs néophytes.

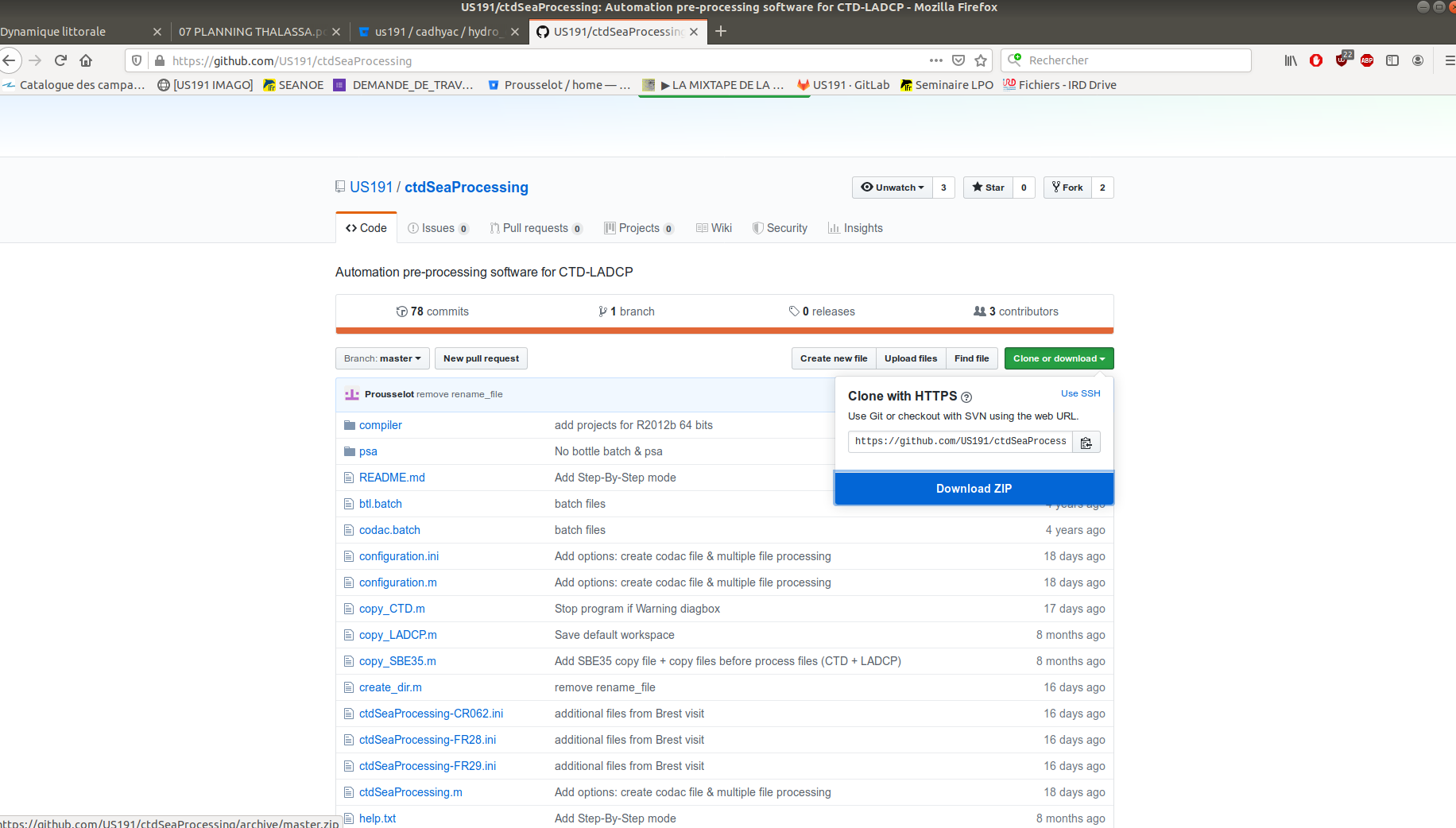
## Matlab

Le dépôt du projet est disponible sous GitHUB :

<https://github.com/US191/ctdSeaProcessing>

### Installation

Télécharger l’archive en format ZIP :



Puis décompresser l’archive dans le dossier désiré.

Si vous avez déjà Linux ou Git d’installé sur votre PC, vous pouvez cloner directement le dépôt en ligne de commande, et basculer sur la branche désirée avec :

git clone <https://github.com/US191/ctdSeaProcessing.git> ctdSeaProcessing

cd ctdSeaProcessing

git checkout master

Le logiciel gzip2 est nécessaire si l’envoi des données en temps réel est demandé pendant la campagne. Ce logiciel permet la compression des fichiers traités. Le dépôt du projet est disponible sous GitHUB :

https://github.com/anandology/gzip2

### Utilisation

En début de campagne, préparer les fichiers .psa, sous /ctdSeaProcessing/psa, avec un éditeur de texte ou avec SBEDataProcessing-Win32selon la configuration de capteurs et les paramètres de traitements désirés.

Créer ensuite le fichier de configuration de campagne, à partir du fichier configuration.ini, avec les différentes informations spécifiques à la campagne :

name\_mission Nom de mission

id\_mission ID de mission

filename\_LADCPM Nom des fichiers issus du LADCP maître (si présent)

filename\_LADCPS Nom des fichiers issus du LADCP slave (si présent)

log\_filename Nom du fichier log

Les options par défaut du logiciel (true ou false):

copy\_CTD Option pour la copie des fichiers CTD depuis le répertoire d’acquisition vers le répertoire de travail de la campagne

copy\_SBE35 Option pour la copie des fichiers SBE35 (si disponible) depuis le répertoire d’acquisition vers le répertoire de travail de la campagne

copy\_LADCP Option pour la copie des fichiers LADCP (si disponible) depuis le répertoire d’acquisition vers le répertoire de travail de la campagne

process\_CTD Option pour le traitement des fichiers CTD

process\_LADCP Option pour le traitement des fichiers LADCP

process\_PMEL Option pour le traitement des fichiers CTD selon les critères du PMEL (à réaliser pour les profils effectués proches des bouées PIRATA)

create\_CODAC Option pour compresser les fichiers CTD traités pour l’envoi en temps réel (dans le cadre de CORIOLIS)

process\_BTL Option pour le traitement des fichiers BTL (si disponible)

Ainsi que les chemins des répertoires utilisés :

drive Répertoire de travail de la campagne

path\_SEASOFT Répertoire d’acquisition des fichiers CTD

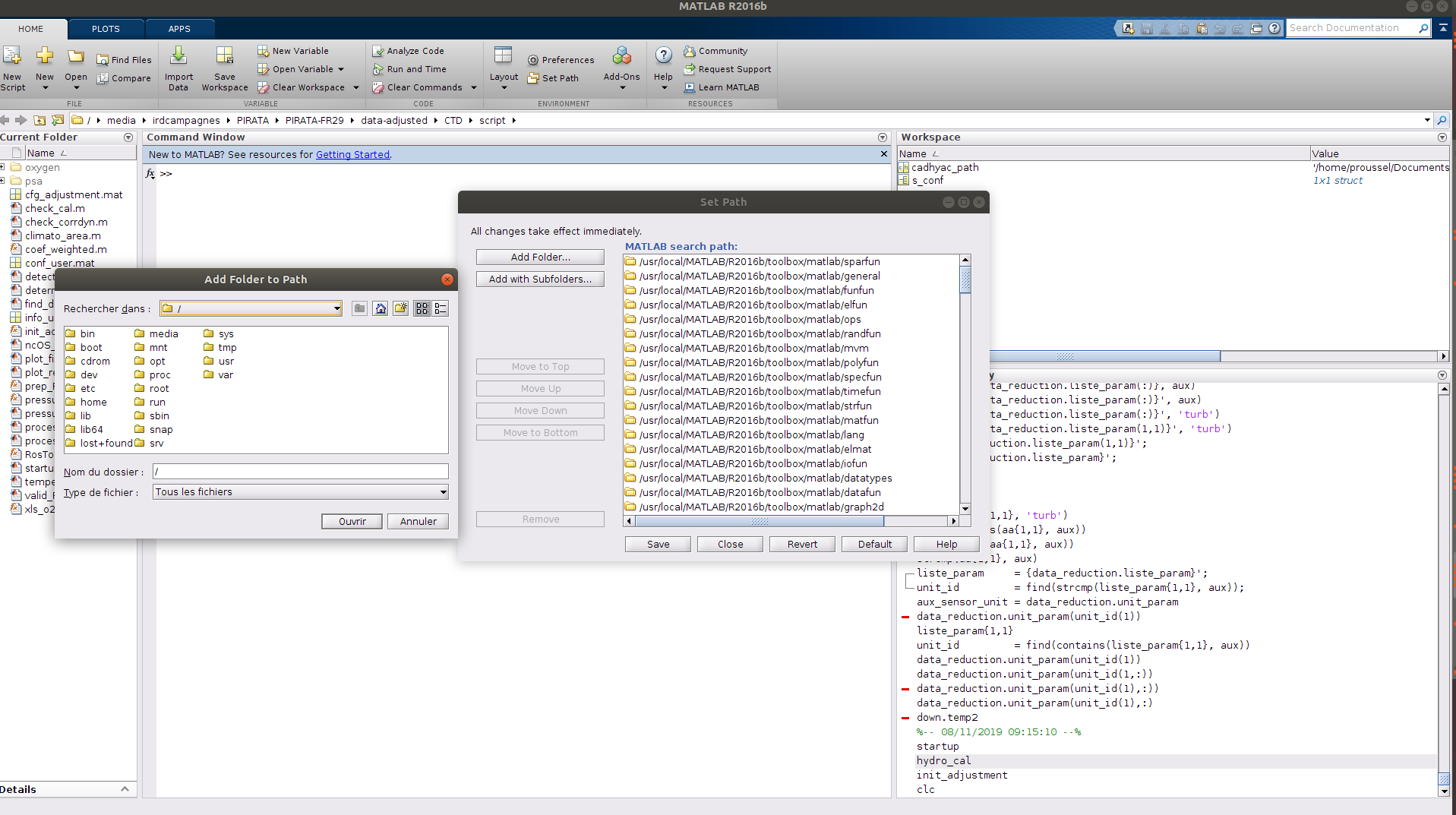
path\_LADCP Répertoire d’acquisition des fichiers LADCP

Puis sauvegarder le fichier de configuration dans un fichier propre à la campagne (ie : configuration\_<cruise>.ini).

Ouvrir ensuite Matlab puis se placer dans le répertoire de ctdSeaProcessing :

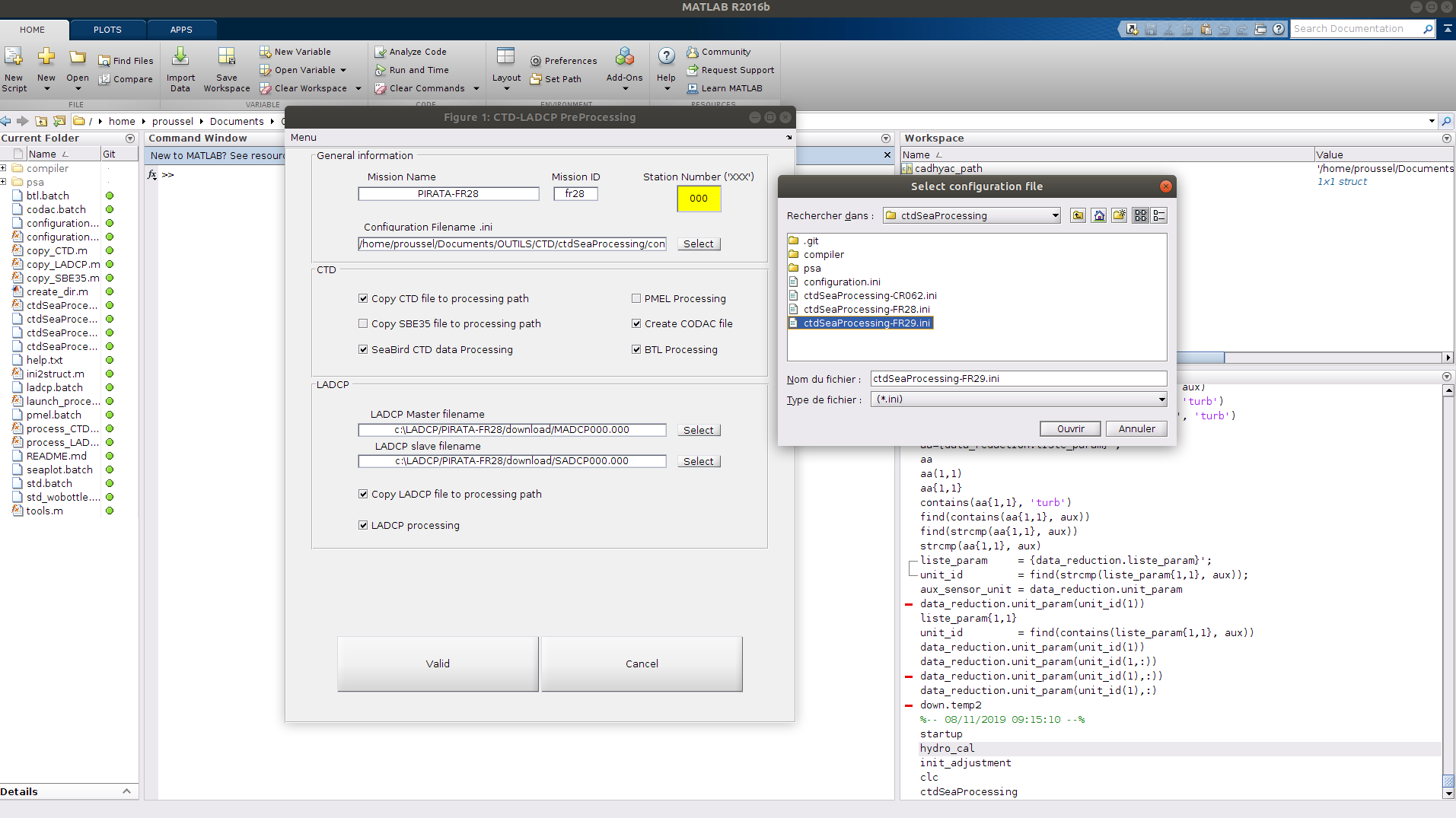
> cd <path>/ctdSeaProcessing

Ou ajouter le chemin dans les path généraux Matlab via Set Path :



Lancer le programme à l’aide de la commande :

> ctdSeaProcessing

Sélectionner le fichier de configuration de la campagne :

Entrer le numéro de station avec trois caractères (‘XXX’). Puis valider pour lancer le processus. (Note : il est possible d’effectuer le processus pour plusieurs profiles successifs en entrant dans le numéro de station ‘XXX – XXX’).

## DOS

Cette version plus ancienne fonctionne sous DOS dans une fenêtre « command.com » et permet de réaliser le traitement par lot. Le traitement par lot est réalisé actuellement par le script Perl process.pl mais il pourrait être facilement remplacé par un script Python, voir un exécutable, simplifiant ici la configuration.

### Installation des programmes et outils

Installer le langage Perl à partir de la distribution ActiveState :

<https://www.activestate.com/products/perl/downloads/>

Depuis cette page, il faudra créer un compte et configurer l’installateur (OS, package additionnel, etc)

Ou plus simple, installer l’outils “state”, avec la commande powershell suivante :

#### Run the following command as an Administrator in your Command Prompt to install it...

powershell "IEX(New-Object Net.WebClient).downloadString('https://platform.activestate.com/dl/cli/install.ps1')"

#### Set this project up locally with the following command...

state activate jgrelet/CTD

Installer également depuis internet les utilitaires bzip2 et gawk, pour Windows, depuis le site GnuWin, binaires d’installation à récupérer sous :

<http://gnuwin32.sourceforge.net/packages/bzip2.htm>

<http://gnuwin32.sourceforge.net/packages/gawk.htm>

Le script de traitement process.pl fait appel à la fin de programme SbeConReport.exe, pour que ce dernier fonctionne, il faut rajouter le chemin de SBEDataProcessing-Win32 dans le path Windows, ainsi que le chemin des utilitaires GnuWin :

Path=C:\Program Files (x86)\Sea-Bird\SBEDataProcessing-Win32;C:\Program Files (x86)\GnuWin32\bin;

### Copie des fichiers bruts Seabird

Copier l’intégralité des fichiers bruts issus du programme d’acquisition Seasave sous :

* data-raw/CTD
* data-processing/CTD/data/raw

Les données copiées dans le répertoire d’archivage data-raw ne doivent jamais être modifiées, par contre, celle qui se trouvent dans le répertoire de travail, sous data-processing/CTD/data/raw peuvent être modifiées et corrigées si nécessaire.

### Utilisation

**Remarque** : Les programmes Seabird utilisés par la chaine de traitement Sbe-Processing ne fonctionnent que sous Windows, il n’est pas possible de lancer le traitement depuis un shell bash.

Editer le script process.pl et modifier/adapter le chemin d’accés au données au début du script :

# modify the variables for the cruise

my $cruise = 'PIRATA-BR18';

my $fileName = "PIRA\_%03d";

# TODOS: get path from filename

my $drive = "C:\\PIRATA";

Avant de commencer le traitement, vérifions que l’interpréteur Perl est bien installé, pour cela ouvrir une console DOS (cmd) puis se placer dans le répertoire :

<cruise>/data-processing/CTD et entrer la commande suivante :

> perl process.pl

usage: perl process.pl [options] first\_station [last\_station|"#w"]

Options:

--help Display this help message

--debug display command, don't process

--btl process sample

--codac process realtime data for CODAC

--pmel process realtime data for PMEL

--ladcp process LADCP data

--std|ctd process Seabird cast

--plot process plots

--report process Seabird report

--all process all steps, default

example: perl process.pl 1

example: perl d:\PIRATA-BR\2018\_PIRBR\_18\data-processing\CTD\process.pl 01

example: perl d:\PIRATA-BR\data-processing\CTD\process.pl 01 09 –std

Afin que la console DOS s’ouvre directement dans le bon répertoire, entrer cmd » dans la barre de recherche, puis cliquer sur « ouvrir l’emplacement d’origine ». Faire un raccourci du programme sur le bureau, puis cliquer avec le bouton droit de la souris sur son icone et dans l’onglet « Raccourci», entrer le chemin d’accès au donnés en modifiant le champ « Démarrer dans ».

Tester le fonctionnement du programme, ex pour la station 1 :

perl process.pl 1

Si vous avez changé la configuration de la CTD, supprimé/rajouté un capteur, les programmes de traitement de la chaine Sbe-processing, vont s’arrêter dès la lecture de l’entête du fichier de données, car la configuration présente dans le fichier \*.psa ne correspond pas avec celle trouvée dans l’entête.

Dans ce cas, introduire la séquence #w dans la ligne de commande, qui aura pour effet de stopper l’exécution avec le chargement du fichier de configuration psa, l’utilisateur pouvant alors modifier la configuration, puis la sauvegarder pour le prochain traitement, et lancer l’exécution du programme.

perl process.pl 1 #w

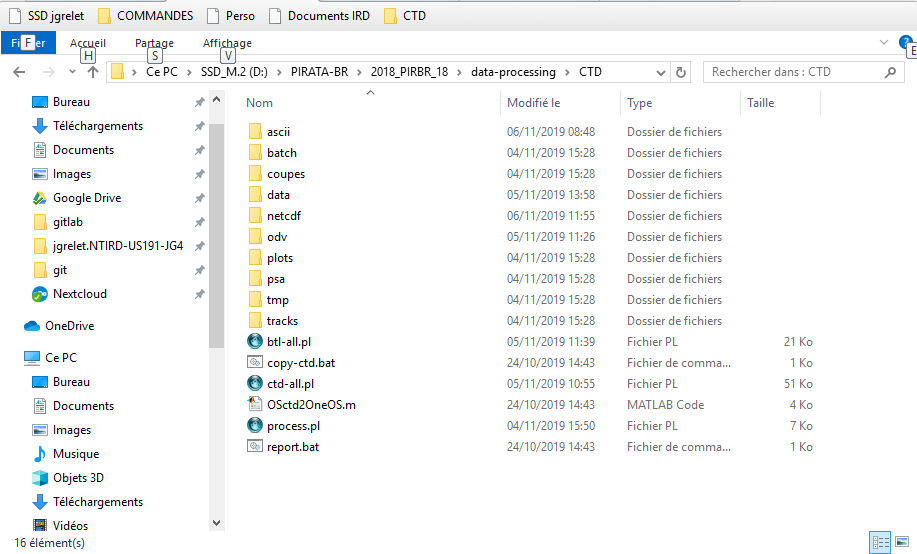


Figura 4: structure du répertoire CTD

# Traitements globaux

## Installer le sous-système Windows pour Linux WSL1

Dans cette documentation, nous décrirons uniquement l’installation et la configuration de WSL1 sous Windows 10.

Suivre pas à pas les instructions du Guide d’installation du sous-système Windows pour Linux pour Windows 10 :

<https://docs.microsoft.com/fr-fr/windows/wsl/install-win10>

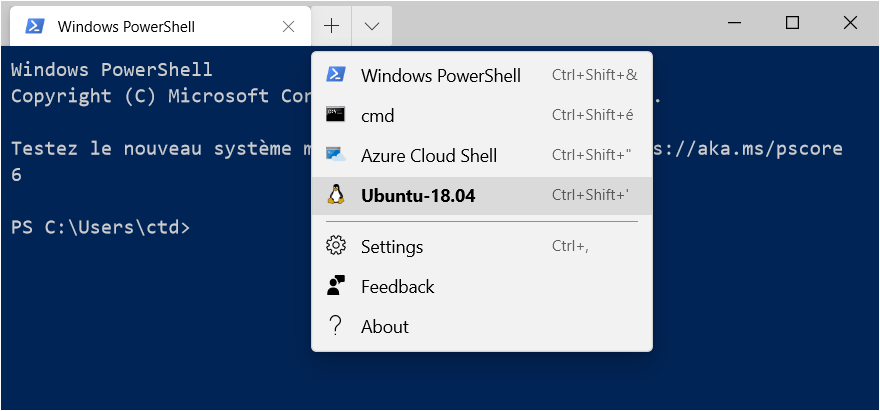
Avant d’installer des distributions Linux pour WSL, vous devez vous assurer que la fonctionnalité facultative « Sous-système Windows pour Linux » est activée :

1. Ouvrez PowerShell en tant **qu’administrateur** et exécutez :

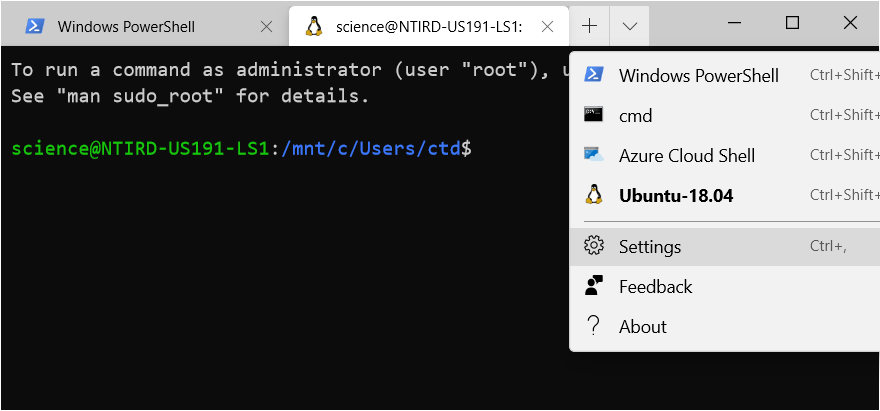
PowerShell

* Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux
* Redémarrez votre ordinateur lorsque vous y êtes invité.

1. Installer le nouveau terminal Windows depuis le Microsoft Store. La version actuelle est baptisée « Terminal Preview »
2. Télécharger et installer à partir du Microsoft Store la distribution Ubuntu 18.04 LTS
3. Une fois la session Linux (Ubuntu) démarrée et le compte utilisateur créé, vous pouvez lancer le Terminal Windows. Ce dernier ouvre par défaut une session PowerShell. Dans le menu (flèche vers le bas), sélectionner Ubuntu-18.04



Pour ouvrir une session Ubuntu par défaut, il faut éditer le fichier de configuration (setting) et remplacer le guid par défaut du Powershell par celui d’Ubuntu. Voir « *Editing Windows Terminal JSON Settings »* sur la page [Github](https://github.com/microsoft/terminal/blob/master/doc/user-docs/UsingJsonSettings.md) du projet.



1. mettre à jour la distribution :

sudo apt-get update

sudo apt-get upgrade

1. Installer Xfce (optionnel) si vous désirez faire de l’affichage graphique :

Installer uniquement l'environnement Xfce avec la commande  :  
  
sudo apt-get xfce4 vim-gtk  
  
Editer le fichier ~/.bashrc et ajouter la variable d'environnement :

export DISPLAY:0.0

Recharger le fichier ~/.bashrc:

. ~/.bashrc

1. Installer un serveur X sur le PC pour afficher les logiciels lancés sous X :

Pour pouvoir lancer des applications graphiques, il faut installer un serveur X externe pour Windows, comme [Xming](https://en.wikipedia.org/wiki/Xming) ou [VcXsrv](https://sourceforge.net/projects/vcxsrv/).   
Voir les détails sous : <https://doc.ubuntu-fr.org/wsl>  
Configurer l’affichage sur localhost :0.0

1. Tester l’affichage X sous Windows avec les programmes xeyes ou gvim
2. Installer ensuite les paquets suivants qui sont nécessaires pour les traitements ultérieurs :

sudo apt install netcdf-bin libnetcdf-dev perl-doc libswitch-perl libdate-manip-perl libxml-libxml-perl libconfig-tiny-perl pdl libpdl-netcdf-perl

Installer le module Oceano Perl de l’US191 IMAGO. Ce module se trouve dans le répertoire local/src du dépôt CRDAP. Sinon, le récupérer avec git sur le dépôt oceano comme suit :

mkdir –p ~/src/perl/modules

cd ~/src/perl/modules

git clone https://git.outils-is.ird.fr/grelet/oceano.git

cd oceano/lib/perl/Oceano

perl Makefile.PL

make

sudo make install

Editer votre fichier ~/.bashrc et y copier les lignes suivantes à la fin, en faisant pointer la variable DRIVE vers le répertoire de la campagne à traiter :

# brazil cruise on hard disk

export DRIVE=/mnt/c/PIRATA

# french cruise on external or share drive

# export DRIVE=/mnt/m/campagnes/PIRATA

# if local mount doesn’t exist (see df –k), execute following commands

# sudo mkdir /mnt/m

$ sudo mount -t drvfs M: /mnt/m

# Cruise main directory name

export CRUISE=PIRATA\_18

#export CRUISE=PIRATA-FR30

echo "Trying to source ${DRIVE}/${CRUISE}/local/etc/skel/.bashrc.${CRUISE}"

if [ -f ${DRIVE}/${CRUISE}/local/etc/skel/.bashrc.${CRUISE} ]; then

. ${DRIVE}/${CRUISE}/local/etc/skel/.bashrc.${CRUISE}

echo "Yes, seems good !!!"

else

echo "Can't source file !!! check your network, hard drive and/or ENV variables !!!"

fi

Ressourcer le fichier ~/.bashrc :

. ~/.bashrc

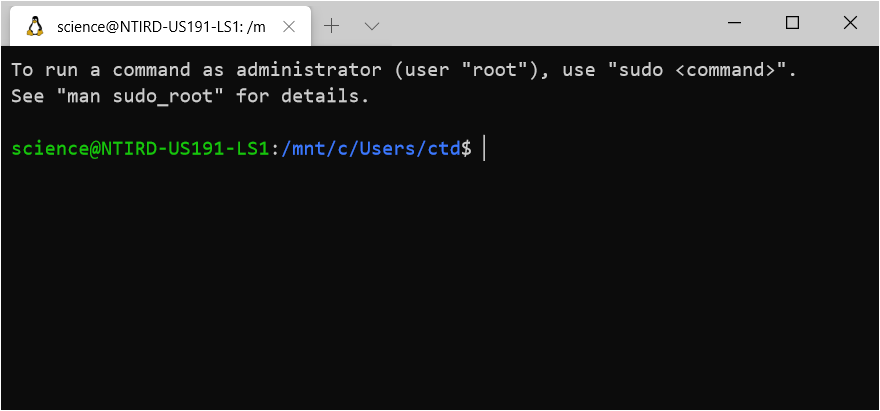


Figura 5 : prompt shell bien formé

Si vous voulez réaliser les traitements depuis un partage réseau, il faut monter ce partage au préalable sous /mnt. Par exemple, pour le partage M :

$ sudo mkdir /mnt/m  
$ sudo mount -t drvfs M: /mnt/m

$ sudo mkdir /mnt/data  
$ sudo mount -t drvfs O: /mnt/data

Python :

sudo ln -s /usr/bin/python3 /usr/bin/python

Installation des modules python avec pip:

sudo ln –s /usr/bin/python3 /usr/bin/python

sudo apt install python3-pip

pip3 install netCDF4

pip3 install xarray

pip3 install matplotlib

Installation des modules python avec miniconda:

mkdir ~/miniconda

cd ~/miniconda

wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86\_64.sh

chmod +x Miniconda3-latest-Linux-x86\_64.sh

bash Miniconda3-latest-Linux-x86\_64.sh

conda install -c conda-forge cartopy

. .bashrc

Voir <https://doc.ubuntu-fr.org/miniconda>

wsl-2: voir :

https://docs.microsoft.com/fr-fr/windows/wsl/wsl2-install

## Configuration

Modifier le fichier de configuration config.ini en adaptant les méta-data aux caractéristiques de la campagne, section « cruise » et à la configuration de la CTD dans la section « ctd ».

A compléter…

[global]

author = jgrelet IRD nov 2019 PIRATA-BR18 cruise

debug = 0

echo = 1

[cruise]

cycle\_mesure = PIRATA-BR18

plateforme = VITAL DE OLIVEIRA

callsign = XXXX

imo = 9070307

mmsi = 227297000

context = PIRATA

timezone = GMT

format\_date = DMY

processing\_code= 1A

begin\_date = 01/09/2018

end\_date = 04/10/2018

institute = INPE

pi = BRUTO

creator = Jacques.Grelet@ird.fr

[xml]

# dtd = local or public

dtd = public

dtdLocalPath = /m/PIRATA-BR18

#encoding = ISO-8859-1

encoding = UTF-8

[ctd]

cruisePrefix = PIRA\_

# profile name : cspxxxyy

stationPrefixLength = 3

acquisitionSoftware = SEASAVE

acquisitionVersion = 7.22.3

processingSoftware = SBE-PROCESSING

processingVersion = 7.22.3

type = SBE911+

sn = 09P-xxxx

title\_summary = CTD profiles processed during PIRATA-BR18 cruise

comment = CTD profiles with pre-cruise coefficients, temperature, salinity and oxygen from primary and secondary sensors

split = ETDD,2,PRES,3,DEPH,4,TEMP,5,CNDC,6,TUR3,10,FLU2,9,DOX2,11,PSAL,12,DENS,13,SVEL,14,NAVG,15

header = PRFL PRES DEPH ETDD TEMP PSAL DENS SVEL DOX2 FLU2 TUR3 NAVG

format = %05d %6.1f %6.1f %10.6f %7.4f %7.4f %6.3f %7.2f %6.2f %6.3f %7.4f %4d

odv\_hdr = PRES TEMP PSAL DENS SVEL DOX2 FLU2

odv\_unit = [db] [C] [Psu] [kg/m3] [m/s] [ml/l] [milligram/m3]

[ctd-all]

split = ETDD,2,PRES,3,DEPH,4,TE01,5,TE02,6,CND1,7,CND2,8,DOV1,9,DOV2,10,DVT1,11,DVT2,12,TUR3,13,FLU2,14,FLU3,15,DO12,16,DO22,17,PSA1,18,PSA2,19,DEN1,20,DEN2,21,SVEL,22,NAVG,24

#split = ETDD,2,PRES,3,DEPH,4,TE01,5,TE02,6,CND1,7,CND2,8,DOV1,9,DOV2,10,DVT1,11,DVT2,12,FLU2,13,TUR3,14,DO12,15,DO22,16,NAVG,17,PSA1,18,PSA2,19,DEN1,20,DEN2,21,SVEL,22

header = PRFL PRES DEPH ETDD TE01 TE02 PSA1 PSA2 CND1 CND2 DEN1 DEN2 SVEL FLU2 FLU3 TUR3 DO12 DOV1 DVT1 DO22 DOV2 DVT2 NAVG

format = %05d %6.1f %6.1f %10.6f %7.4f %7.4f %7.4f %7.4f %7.5f %7.5f %6.3f %6.3f %7.2f %6.3f %6.3f %6.2f %7.3f %6.4f %+7.5f %7.7g %6.6g %+7.7g %4d

#format = %05d %4d %6.1f %10.6f %7.4f %7.4f %7.4f %7.4f %7.5f %7.5f %6.3f %6.3f %7.2f %6.3f %6.2f %7.3f %7.3f %6.4f %6.4f %+7.5f %+7.5f %4d

[btl]

type = SBE32 standard 24 Niskin bottles

sn = unknown

title\_summary = Water sample during PIRATA-BR18 cruise with 20 levels

comment = CTD bottles water sampling with temperature, salinity and oxygen from primary sensors

split = BOTL,1,month,2,day,3,year,4,PSA1,5,DO11,6,DO12,7,Potemp090C,8,ETDD,9,PRES,10,DEPH,11,TE01,12,CND1,13,DOV1,14,DOV2,15,TUR3,17,FLU2,16

header = PRFL BOTL PRES DEPH ETDD TE35 TE01 TE02 PSA1 PSA2 CND1 CND2 DO11 DO12 DO21 DO22 DOV1 DOV2 DVT1 DVT2 FLU2 TUR3

format = %05d %2d %6.1f %6.1f %10.6f %7.6g %7.4f %7.4f %7.4f %7.4f %7.5f %7.5f %7.4f %8.4f %7.3f %8.3f %8.5f %8.5f %+7.5f %+7.5f %+6.4f %6.2f

[xbt]

cruisePrefix = br18

stationPrefixLength = 3

acquisitionSoftware = WinMK21

acquisitionVersion = 2.10.1

processingSoftware =

processingVersion =

type = SIPPICAN

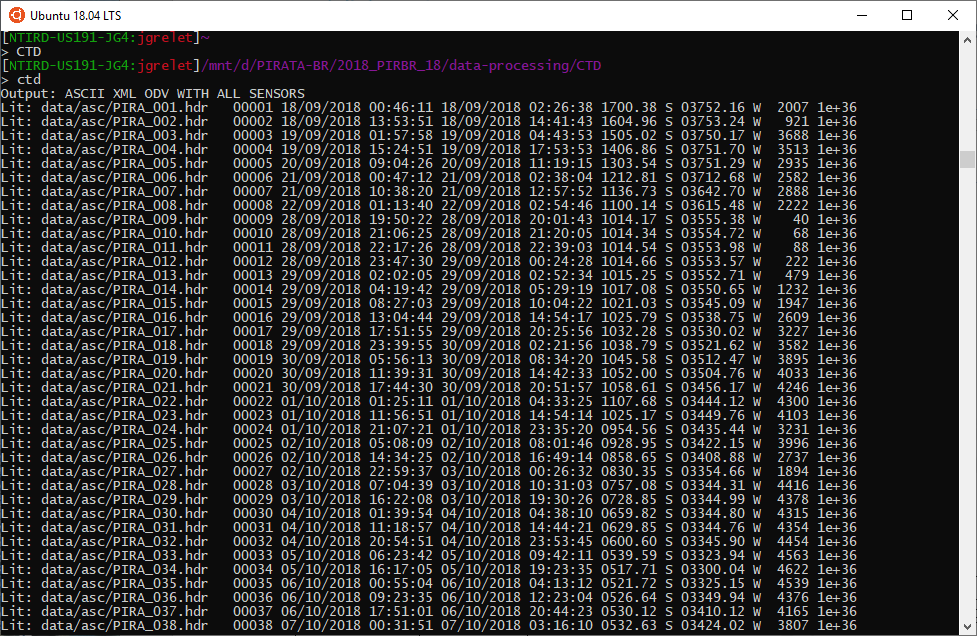
sn = 01150

title\_summary = XBT profiles processed during PIRATA-BR18 cruise

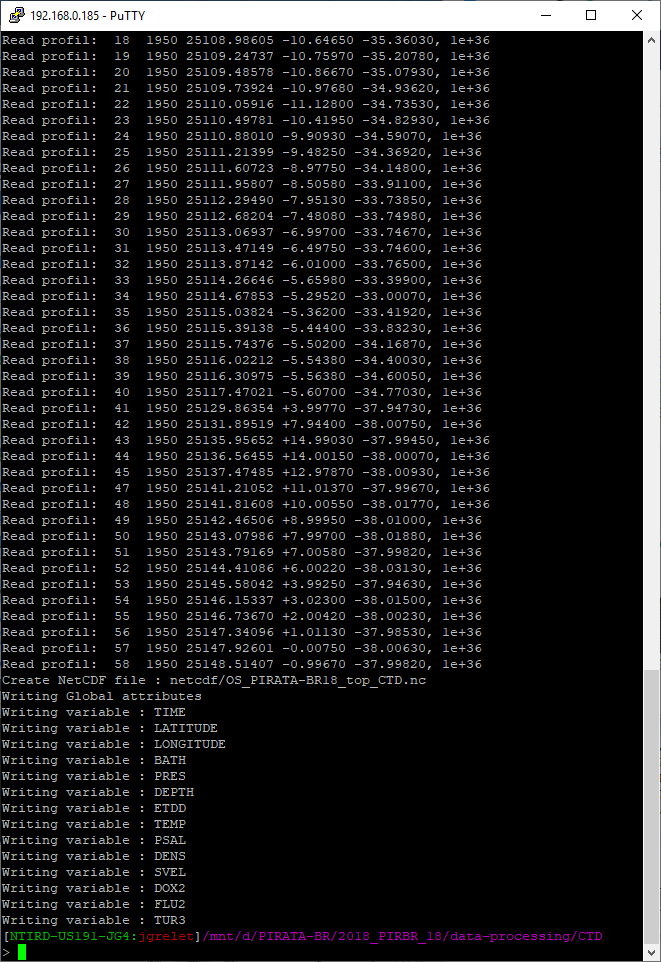
comment = Extract from .edf files

## Utilisation

### Génération des fichiers ASCII d’entêtes, data et XML



### Génération du fichier NetCDF



## Automatisation

### Scripts de copie

Créer un lien symbolique de /mnt/campagnes vers /mnt/m

$ sudo ln –s /mnt/m /mnt/campagnes

Copier, adapter ou créer le script de copie des fichiers au travers du réseau sous local/sbin/synchro.sh

#!/bin/bash

#

# Mission PIRATA-FR30 N/O Thalassa mars 2020 J.Grelet

# script de synchronisation des données acquises par TECHSAS sur mission data-raw et data-processing

# repertoire source

export SOURCE=/mnt/data

# disque de destination

export DRIVE=/m

# repertoire de destination

export DEST=$DRIVE/PIRATA-FR30

# nom utilise par genavir a bord pour la campagne

export CRUISE=PIRATAFR30

echo ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"

echo "Debut de synchro : `/bin/date +%d/%m/%Y\_%H:%M:%S`"

echo ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>" sync

echo " "

# copie SADCP vers data-raw et data-processing

echo "# copie SADCP vers data-raw et data-processing"

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/EQUIPEMENTS/OS38/DONNEES/\* $DEST/data-raw/SADCP/OS38

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/EQUIPEMENTS/OS38/ANNEXES/\* $DEST/data-raw/SADCP/OS38/ANNEXES

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/EQUIPEMENTS/OS38/DONNEES/\*.[L-S]TA $DEST/data-processing/SADCP/OS38/data

# OS150

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/EQUIPEMENTS/OS150/DONNEES/\* $DEST/data-raw/SADCP/OS150

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/EQUIPEMENTS/OS150/ANNEXES/\* $DEST/data-raw/SADCP/OS150/ANNEXES

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/EQUIPEMENTS/OS150/DONNEES/\*.[L-S]TA $DEST/data-processing/SADCP/OS150/data

# LOCH RDI DVL600

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/EQUIPEMENTS/LOCH\_RDI/DONNEES/\* $DEST/data-raw/SADCP/DVL600

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/EQUIPEMENTS/LOCH\_RDI/ANNEXES/\* $DEST/data-raw/SADCP/DVL600/ANNEXES

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/EQUIPEMENTS/LOCH\_RDI/DONNEES/\*.[L-S]TA $DEST/data-processing/SADCP/DVL600/data

# copie TECHSAS ARCHIV\_NETCDF vers data-raw

echo "# copie TECHSAS ARCHIV\_NETCDF vers data-raw"

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/ARCHIV\_NETCDF/DONNEES/THS/\*.ths $DEST/data-raw/TECHSAS/ARCHIV\_NETCDF/THS

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/ARCHIV\_NETCDF/DONNEES/NAV/\*.nav $DEST/data-raw/TECHSAS/ARCHIV\_NETCDF/NAV

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/ARCHIV\_NETCDF/DONNEES/GPS/\*.gps $DEST/data-raw/TECHSAS/ARCHIV\_NETCDF/GPS

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/ARCHIV\_NETCDF/DONNEES/FBOX/\*.fbox $DEST/data-raw/TECHSAS/ARCHIV\_NETCDF/FBOX

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/ARCHIV\_NETCDF/DONNEES/MET/\*.met $DEST/data-raw/TECHSAS/ARCHIV\_NETCDF/MET

# copie TECHSAS ARCHIV\_NMEA vers data-raw et data-processing

echo "# copie TECHSAS ARCHIV\_NMEA vers data-raw et data-processing"

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/ARCHIV\_NMEA/DONNEES/meteo/\*.met $DEST/data-raw/TECHSAS/ARCHIV\_NMEA/METEO

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/ARCHIV\_NMEA/DONNEES/COLCOR/\*.COLCOR $DEST/data-raw/TECHSAS/ARCHIV\_NMEA/COLCOR

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/ARCHIV\_NMEA/DONNEES/COLCOR/\*.COLCOR $DEST/data-raw/THERMO

# bug dans la numerotation des fichiers COLOCOR par TECHSAS lors d'un arret de l'acquisition

# les fichiers sont copies dans data/origin et renomes dans data pour respecter l'ordre d'acquisition

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/ARCHIV\_NMEA/DONNEES/COLCOR/\*.COLCOR $DEST/data-processing/THERMO/data/origin

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/ARCHIV\_NMEA/DONNEES/thsal/\*.sal $DEST/data-raw/TECHSAS/ARCHIV\_NMEA/SBE21

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/ARCHIV\_NMEA/DONNEES/FYBOX/\*.FYBOX $DEST/data-raw/TECHSAS/ARCHIV\_NMEA/FYBOX

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/ARCHIV\_NMEA/DONNEES/GILLA/\*.gill $DEST/data-raw/TECHSAS/ARCHIV\_NMEA/GILLA

/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/ARCHIV\_NMEA/DONNEES/.sonde18/\*.snd $DEST/data-raw/TECHSAS/ARCHIV\_NMEA/SONDE18

# copie CASINO vers data-raw et data-processing a partir de M:\PIRATAFR26\DONNEES\_BORD\CASINO

echo "# copie CASINO vers data-raw et data-processing"

#/bin/cp -ruv $SOURCE/DONNEES/$CRUISE/CASINO/DONNEES/\* $DEST/data-raw/CASINO

#/bin/cp -ruv $SOURCE/DONNEES/$CRUISE/CASINO/DONNEES/CSV/\*.csv $DEST/data-processing/CASINO/data

/bin/cp -ruv $DRIVE/CASINO/\* $DEST/data-raw/CASINO

/bin/cp -ruv $DRIVE/CASINO/CSV/\*.csv $DEST/data-processing/CASINO/data

# copie XBT .edf vers data-raw et data-processing

echo "# copie XBT .edf de data-raw et data-processing"

#/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/CELERITE/\* $DEST/data-raw/CELERITE

#/bin/cp -rupv $SOURCE/DONNEES/$CRUISE/CELERITE/\*.edf $DEST/data-processing/CELERITE/DATA

/bin/cp -rupv $DRIVE/CELERITE/\* $DEST/data-raw/CELERITE

/bin/cp -rupv $DRIVE/CELERITE/\*.edf $DEST/data-processing/CELERITE/DATA

echo " "

echo ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"

echo "Fin de synchro : `/bin/date +%d/%m/%Y\_%H:%M:%S`"

echo ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"

### Scripts de traitements

Copier, adapter ou créer le script de traitement des fichiers sous local/sbin/process-all.sh

#!/bin/bash

echo ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"

echo "Begin process: `/bin/date +%d/%m/%Y\_%H:%M:%S`"

echo ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"

echo " "

shopt -s expand\_aliases

export HOME=/home/science

export DRIVE=/mnt/campagnes

export CRUISE=PIRATA-FR30

echo "Trying to source ${DRIVE}/${CRUISE}/local/etc/skel/.bashrc.${CRUISE}"

if [ -f ${DRIVE}/${CRUISE}/local/etc/skel/.bashrc.${CRUISE} ]; then

. ${DRIVE}/${CRUISE}/local/etc/skel/.bashrc.${CRUISE}

echo "Yes, seems good !!!"

else

echo "Can't source file !!! check your network, hard drive and/or ENV variables !!!"

fi

# check alias for debug

#alias

echo ""

echo "CTD processing:"

echo "---------------"

CTD

ctd

ctdnc

btl

btlnc

echo ""

echo "TSG processing:"

echo "---------------"

TSG

tsg

tsgnc

ctd-tsg

echo ""

echo "XBT processing:"

echo "---------------"

XBT

xbt

xbtnc

python plot\_profile.py

echo ""

echo "LADCP processing:"

echo "-----------------"

LADCP

ladcp

ladcpnc

echo ""

echo "CASINO processing:"

echo "------------------"

CASINO

casino

casinonc

casinosndnc

casinotsgnc

casinofboxnc

echo ""

echo "SADCP cat files:"

echo "----------------"

SADCP

cat\_all\_files.sh

echo ""

echo "GoogleEarth cruisetrack processing:"

echo "------------------------------------"

CTD

cd tracks

$LOCAL/sbin/linux/cruiseTrack2kml-linux-amd64 -config local.toml -output pirata-fr30-local.kml

$LOCAL/sbin/linux/cruiseTrack2kml-linux-amd64 -config config.toml -output pirata-fr30.kml

echo " "

echo ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"

echo "End of process : `/bin/date +%d/%m/%Y\_%H:%M:%S`"

echo ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"

### Crontab

Editer la crontab avec la commande crontab –e et ajouter les lignes suivantes :

SHELL=/bin/sh

CRUISE=PIRATA-FR30

5 7,9,11,17,22 \* \* \* /mnt/campagnes/${CRUISE}/local/sbin/synchro.sh > /mnt/campagnes/${CRUISE}/local/logs/synchro.log 2>&1

30 7,9,11,17,22 \* \* \* /mnt/campagnes/${CRUISE}/local/sbin/process-all.sh > /mnt/campagnes/${CRUISE}/local/logs/process.log 2>&

Le déroulement de la copie et du traitement est enregistré dans des fichiers de log

# Suivi des versions de ce document

|  |  |  |  |
| --- | --- | --- | --- |
| **Rédacteur** | | **Approbateur** | |
| Nom : | J. Grelet | Nom : | J. Grelet |
| Fonction : | Responsable laboratoire physique de Brest | Fonction : | Responsable laboratoire physique de Brest |

|  |  |  |
| --- | --- | --- |
| **Date** | **Version** | **Commentaires et modifications** |
| 11/01/2019 | 1 | Version initiale rédigée dans le cadre de ma MLD à Recife |
|  |  |  |
|  |  |  |

|  |  |
| --- | --- |
| **Relecteur** | **Date** |
|  |  |