using Distributed

npps3 = 8

using LibPQ

using HDF5

using CSV

using DataFrames

host="localhost"

port="5432"

db="nemo"

conn = LibPQ.Connection("dbname=$db user=$user password=$pwd sslmode=disable")

#fbt="ArcticaM1\_202309191645\_Brightness\_Temperature\_09.h5"

#fgt="ArcticaM1\_202309191645\_LatLon\_4km.h5"

fbt="fdata.h5"

fgt="fgeo.h5"

fid = h5open(fbt, "r")

fgd = h5open(fgt, "r")

g = fid["Data"]

resol = g["resolution\_4km"]

dset = resol["Brightness\_Temperature\_09"]

as = 100.

A = read(dset)

glat = fgd["Geolocation/resolution\_4km/Latitude"]

lat = read(glat)/as

glon = fgd["Geolocation/resolution\_4km/Longitude"]

lon = read(glon)/as

radtemp91 = A /as

close(fid)

close(fgd)

# чтение данных из файлов данных и геопривязки

tfsv="tcurdate.txt"

tffs = CSV.File(tfsv,header=["curdate"]) |> DataFrame

tf = tffs[1,:1]

# чтение времени из файла – строка вида "20230919 1645 "

# будет записана в базу в поле “tdate”

addprocs(npps3)

@everywhere using DataFrames

@everywhere using IterTools

@everywhere using LibPQ

@everywhere using DataFramesMeta

@everywhere include("corefunc.jl")

# подключение файла с необходимыми функциями

R = [ @spawnat nidf Splitar3(radtemp91,lat,lon,nidf-1,tf) for nidf=2:9 ]

Ss = 0

for nidfs = 1:8

global Ss = Ss + fetch(R[nidfs])

end

# запуск главной функции в параллельном режиме

# финализация запущенных дочерних процессов

#//////////////////

@time insert\_by\_copy!(conn, "armhb51", dff1)

# загрузка данных в базу