

Apèndix 4: Descodificador d'arxius BUFR

1.-BUFRdecode_ADPSfc.tar	3
2.-GRABBUFR.f	5
3.-BUFRDECODE.f	8

1.-BUFRdecode_ADPsfc.tar

Per passar els arxius BUFR a arxius de text s'ha fet servir el descodificador que et proporciona la base de dades (<http://dss.ucar.edu/datasets/ds461.0/software/>): BUFRdecode_ADPsfc.tar. Aquest programa està disponible a la pàgina web de forma gratuïta, i consta dels següents mòduls: grabbufr, bufradpsfc, bufrsfcsfp i dumpbufr. De forma breu es realitzarà una descripció de cada un d'aquests programes:

- Grabbufr. Passa els arxius de big-endian a little-endian. Aquests formats es diferencien per l'ordre d'emmagatzemar els múltiples bytes d'una dada. El format és funció del tipus de processador utilitzat. Els little-endian (ex. Intel) emmagatzemen el byte de menor pes a la posició de memòria més baixa i el byte de major pes a la més alta (de més petit a més gran). En canvi els big-endian (ex. Apple) ho fan de forma contrària, el byte de major pes anirà a la direcció de memòria més baixa i el més petit a la més alta (de més gran a més petit). Un cop compilat i executat el programa es necessari introduir la mida de l'arxiu.
- Bufradpsfc. Descodifica els arxius BUFR a arxius de text. Aquest programa és vàlid per observacions terrestres, i no per les oceàniques. L'estructura dels arxius de les observacions terrestres és diferent a la de les oceàniques, la qual cosa fa que sigui necessari fer servir programes diferent. Tot i això, els descodificadors són molt semblants. Abans de compilar el programa, al codi font s'han d'especificar les variables que es volen extreure dels arxius BUFR.
- Bufrsfcsfp. Descodifica els arxius BUFR a arxius de text. Aquest és el programa per les observacions oceàniques. De la mateixa manera que al bufradpsfc, s'han de introduir les variables desitjades al codi font, abans de realitzar la compilació. Aquest programa no s'ha utilitzat durant el projecte, però la seva execució és molt semblant a la del bufradpsfc.
- Dumpbufr. Realitza un bolc de totes les dades de l'arxiu BUFR. És independent quin tipus d'arxiu s'utilitzi, terrestre o oceànic. Aquest programa no es fa servir durant el processat de dades, per tant no es coneix el seu funcionament de forma tan detallada.

Un cop descrits els diferents mòduls del descodificador, veurem com s'estructura i quin seguit de passos s'han de realitzar per executar-lo de forma correcta. Està estructurat en un seguit de carpetes, com es veurà a continuació:

- `bufrobs`. Carpeta a on s'han de col·locar els arxius BUFR que es volen descodificar.
- `doc`. Hi ha arxius amb les taules necessàries per l'escriptura dels arxius de text.
- `exe`. Carpeta a on es creen els executables un cop realitzada la compilació dels programes.
- `grabbufr`. Carpeta a on es troba el mòdul de conversió d'arxius big-endian a little-endian. Consta del codi font del programa, un script per realitzar la instal·lació i un arxiu d'ajuda.
- `install`. A la carpeta hi ha dos scripts per poder realitzar la instal·lació dels descodificadors (`bufradpsfc`, `bufrsfeshp` i `dumpbufr`).
- `lib`. Aquí és a on s'emmagatzema tota la informació necessària per poder realitzar la descodificació. Són un conjunt d'arxius amb codi font que són cridats per el programa descodificador i contenen tota l'estructura, el funcionament i les taules dels arxius BUFR.
- `src`. A aquesta carpeta és a on hi ha el codi font dels programes `bufradpsfc`, `bufrsfeshp` i `dumpbufr`. Com s'ha comentat anteriorment, aquests programes fan servir els arxius de la carpeta `lib`, per realitzar la descodificació. Són aquests es arxius que s'han de modificar per seleccionar les variables d'interès.
- `textobs`. Finalment aquesta és la carpeta a on es guarden els arxius de text un cop realitzada la descodificació.

Finalment cal dir que per realitzar la descodificació s'ha fet servir una màquina Linux amb el sistema operatiu Debian 4.0 instal·lat. Els programes estan escrits en dos llenguatges de programació diferents: Fortran i C. Els compiladors utilitzats són `g95` i `gcc`, tal i com recomanava la base de dades. Per la realització dels scripts s'ha utilitzat el llenguatge `csh` (C shell).

2.-GRABBUFR.f

Aquest programa passa els arxius de big-endian a little-endian. El programa original requeria llegir per pantalla el nom de l'arxiu d'entrada, la seva mida (en bytes) i el nom de l'arxiu de sortida. Això suposa un problema si s'han de processar moltes dades, per tant, per resoldre'l s'ha modificat el programa fent llegir d'un arxiu totes les variables necessàries, tal i com es pot observar a continuació:

Programa inicial:

```
program grabbufr
  character,allocatable :: cbuf(:)
  CHARACTER(len=80) :: infile,outfile
  character(len=4) :: bufr='BUFR',ctemp,csec0
  INTEGER(4)      narg,iargc,JSTAT(100)
  integer findbufr
  character*1 byte(8)
  data i1/11/,i2/51/,newed/2/
  call wrdlen
```

C Llegeix per pantalla l'arxiu d'entrada i l'arxiu de sortida, i els assigna a infile i outfile, respectivament, mitjançant l'ordre "call getarg".

```
NARG=IARGC()
IF(NARG.NE.2) THEN
  PRINT *,'grabbufr: Incorrect usage'
  PRINT *,'Usage: grabbufr inputBUFRfile ouputBUFRfile'
  CALL EXIT(2)
ENDIF
call getarg(1,infile)
infile = TRIM(infile)//CHAR(0)
call getarg(2,outfile)
outfile = TRIM(outfile)//CHAR(0)
```

C Llegeix per pantalla la mida de l'arxiu d'entrada. Aquesta dada es necessària ja que es fa servir la funció "allocate" (dimensionat dinàmic), i cal determinar l'espai de memòria necessari.

```
READ(5,*,END=900,ERR=900) KBYTES
PRINT *,'NUMBER OF BYTES IN INPUT BUFR FILE   = ',KBYTES

allocate(cbuf(kbytes),stat=istat)
IF (istat.ne.0) THEN
    PRINT *,' ERROR ',kbytes,' bytes to read in file ',infile
    CALL EXIT(4)
ENDIF
...
end
```

Programa modificat

```
program grabbufr
```

C Es defineix el paràmetre n=250, que representa el nombre màxim d'arxius que es processaran (31 dies x 8 arxius = 248). El processat s'ha realitzat de forma mensual.

```
parameter (n=250)
character,allocatable :: cbuf(:)
CHARACTER(len=80) :: infile,outfile
character(len=4) :: bufr='BUFR',ctemp,csec0
integer findbufr
character*1 byte(8)
dimension YTES(n)
data i1/11/,i2/51/,newed/2/

call wrdlen
```

C S'obre l'arxiu a on trobem la mida de l'arxiu d'entrada, el nom de l'arxiu d'entrada i el nom de l'arxiu de sortida, en aquest ordre.

```
open(1,FILE='dades.txt',STATUS='old')
```

C Aquest bucle ens permet executar el programa per a cada arxiu.

```
do k=1,n
```

C Es llegeix la línia k de l'arxiu "dades.txt", i assignem a cada variable el seu valor corresponent.

```
read(1,*) YTES(k),infile, outfile
infile = TRIM(infile)//CHAR(0)
outfile = TRIM(outfile)//CHAR(0)
KBYTES=BYTES(k)

PRINT *,'NUMBER OF BYTES IN INPUT BUFR FILE    = ',KBYTES

allocate(cbuf(kbytes),stat=istat)
IF (istat.ne.0) THEN
  PRINT *,' ERROR ',kbytes,' bytes to read in file ',infile
  CALL EXIT(4)
ENDIF
...
```

C Al final del programa, abans de tancar el bucle, es necessari esborrar la mida de la memòria per tal de reassignar el valor de l'arxiu següent. Després tanquem el bucle i el programa.

```
deallocate(cbuf)
enddo
```

end

Un cop explicat les modificacions realitzades al programa GRABBUFR, a continuació podem veure com es l'estructura de l'arxiu "dades.txt":

10042192	gdas.adpsfc.t00z.20070401.bufr	adpsfc.2007040100.bufr
10008776	gdas.adpsfc.t00z.20070402.bufr	adpsfc.2007040200.bufr
10023824	gdas.adpsfc.t00z.20070403.bufr	adpsfc.2007040300.bufr

Per conèixer la mida dels arxius s'ha fet servir la comanda *ls -l*.

3.-BUFRDECODE.f

Aquest és el programa per realitzar la descodificació dels arxius BUFR a arxius de text. Per seleccionar les variables meteorològiques desitjades s'ha de modificar el codi font dels arxius bufradpsfc.f i bufrsfshp.f, per arxius d'observacions terrestres i oceàniques, respectivament. A continuació es mostrarà la part del codi font del descodificador que s'ha de modificar. Com la selecció de les variables és igual per als dos casos únicament es mostra el programa de les observacions terrestre:.

Bufradpsfc.f

```
PARAMETER      ( MXMN = 8 )
PARAMETER      ( MXLV = 86 )
PARAMETER      ( NVAR = 18 )
PARAMETER      ( NSTR = 4 )
COMMON /BITBUF/ MAXBYT,IBIT,IBAY(5000),MAYT(32),MBAY(5000,32)
REAL*8         r8arr ( MXMN, MXLV ), r8arr2(MXMN, MXLV ),
+              r8arr3 ( MXMN, MXLV ), r8arr4(MXMN, MXLV )
PARAMETER      ( MXBF = 16000 )
CHARACTER      cbfmsg*(MXBF),argv*300,date_tag*10,
+              csubset*8, inf*200, outstg*200,fout*30
```

C "ostr(NSTR)" són les cadenes de caràcters a on s'emmagatzema el codi de les variables (MNEMONICS) que es volen descodificar.

```
CHARACTER*80    ostr(NSTR)
INTEGER         ibfmsg ( MXBF / 4 ), ln, code,y,z,i,iargc,n
LOGICAL        msgok
EQUIVALENCE    ( cbfmsg (1:4), ibfmsg (1) )
```

C Aquestes són les variables que es volen extreure (veure BUFR Table B)

```
ostr(1)='WMOB WMOS RPID CLAT CLON'
ostr(2)='SELV YEAR MNTH DAYS HOUR'
ostr(3)='MINU TMDB TMDP REHU'
ostr(4)='PRWE PRES PMSL PRPT'
```

```
n = iargc()
```

C Llegeix el nom del fitxer d'entrada (BUFR) i el nom del fitxer de sortida (txt).

```
call getarg( 1, argv )
inf=argv
call getarg(2,argv)
date_tag=argv

fout= "SFC."//date_tag//'.txt'
open(unit=11,file=inf,form='unformatted')
open(unit=21,file=fout)
```

C A continuació s'escriu la capçalera de l'arxiu. Aquí s'ha de tenir coneixement de l'espai utilitzat per cada variable. A la primera línia es descriu l'espai total utilitzat (en aquest cas 122). A la segona es descriu el format, variables a variable. És

important que la capçalera respecti l'espai de cada variable per així aconseguir les dades en columnes.

```
WRITE (UNIT=21, FMT='("#",122("-"))')
WRITE (UNIT=21, FMT='("#",A7,1X,A4,1X,A6,
+ 1X,A6,1X,A5,3(1X,A6),4(1X,A4),4(1X,A5),2(1X,A8),
+ 1X,A5)') 'DATACAT',
+ 'WMOB','WMOS',
+ 'RPID','LAT','LON','SELV','YEAR','MNTN','DAY',
+ 'HOUR','MIN','TMDB','TMDP','REHU','PRWE','PRES','PMSL',
+ 'PRPT'
WRITE (UNIT=21, FMT='("#",122("-"))')
...
```

C Després de tota la conversió s'han d'escriure els resultats obtinguts. S'ha d'anar amb compte amb els formats utilitzats per cada variable (caràcter, real,...).

```
WRITE (UNIT=outstg, FMT='(A8,1X,F4.1,1X,F6.1,1X,A6,
+ 1X,F5.1,1X,F6.1,1X,F6.1,1X,F6.1,
+ 4(1X,F4.1),4(1X,F5.1),2(1X,F8.1),1X,F5.1)')
+ csubset,
+ (r8arr(i,z), i = 1,5), (r8arr2(i,z), i = 1,5),
+ (r8arr3(i,z), i = 1,4), (r8arr4(i,z), i = 1,4)
...
END
```

A les explicacions del programa es fa referència als MNEMONICS, que són els codis alfabètics que corresponen a cada variable. A la BUFR Table B, a més de les descripcions dels paràmetres de cada les variables, hi trobem els MNEMONICS.