



**UNIVERSIDAD DE BUENOS AIRES
FACULTAD DE INGENIERÍA**

Año 2018 - 2^{do} Cuatrimestre

ALGORITMOS Y PROGRAMACIÓN I (95.11)

TRABAJO PRÁCTICO N°1

INTEGRANTES:

Landrobe, Natalia. <natylandrobe@gmail.com>

Marca Lima, Renán Michael. <michael9763@hotmail.com>

Fraga, Fernando. <fernandofraga98@outlook.com>

1. Enunciado

Implementar un programa ejecutable por línea de comandos que lea de *stdin* una secuencia de datos provenientes de un sensor de un sistema global de navegación por satélites (GNSS) y imprima por *stdout* el contenido que se genera para un archivo de formato típico para almacenar rutas.

Se analizarán los datos recibidos para la interpretación de aquellas líneas que tienen información geográfica, procesarlos e imprimirlos en el formato GPX.

El protocolo de los sensores que se debe interpretar en este trabajo es el protocolo NMEA-0183 donde sólo se interpretarán las frases de tipo GGA.

Los datos de latitud y longitud en NMEA vienen en formato [d]ddmm.mmmm , donde mm.mmmm son los minutos y [d]dd son los grados.

El cálculo de la suma de verificación (checksum) se calcula como el XOR entre todos los caracteres de la sentencia, excluyendo el carácter inicial (\$) donde los caracteres de la suma de verificación que corresponden a los últimos 2 caracteres y el signo * anterior a ellos. Esta suma se utiliza para saber si el mensaje llegó correctamente o debe ser descartado por estar modificado.

Los **argumentos** que debe recibir el programa son:

- ~ **-h, --help**
 - ☐ Muestra una ayuda
- ~ **-n *nombre* , --name *nombre***
 - ☐ Indica el metadato nombre.
- ~ **-f *fecha*, --format *fecha***
 - ☐ Indica la fecha.
- ~ **-Y *año* , --year *año***
 - ☐ Indica el año.
- ~ **-m *mes* , --month *mes***
 - ☐ Indica el mes.
- ~ **-d *día* , --day *día***
 - ☐ Indica el día.

2. Estructura funcional del programa desarrollado

El programa está diseñado para primero, verifique los argumentos, que son ingresados por línea de comandos, con la función `takeArgs`. Esta misma es la que retorna un `status_t` que lo guardamos en una variable llamada `st`. Si `st == ST_HELP` (es decir, el usuario usó, por ejemplo, el argumento `-h`) imprimimos un mensaje de ayuda sobre el programa, si `st ==`

ST_INV, hubo un error en la escritura del argumento, por lo tanto, imprimimos un mensaje de error por stderr.

Luego enviamos el nombre a la función `printMetadata` que imprime en formato XML la metadata, seguimos con la lectura de cada línea recibida, y verificamos con `checkLine` que empiece con \$GPGCC, ya que solo leeremos este tipo de frases. Se corrobora que el checksum sea compatible, y con `cargarStruct` notaremos si la carga de la estructura tipo fecha y tipo data fue satisfactoria, es decir que todos los datos de los miembros sean válidos.

Si los resultados son correctos, a través de `printStruct`, se imprimen los Track points correspondientes que contienen los datos de la latitud y longitud de un punto de la ruta, que van siendo interpretados de la línea GGA.

Cuando el sensor no envia mas lineas a leer, se imprimen los cierres de tags a través de `printTrkc`.

A continuación, mostraremos por medio de una tabla las funciones del programa. para que sea mas practica su lectura

<u>Funciones</u>			
<u>Nombre de la función.</u>	<u>Argumentos recibidos.</u>	<u>Finalidad de la función.</u>	<u>Valor regresado.</u>
printMetadata	Un puntero a char.	Imprimir el tag Metadata en el formato basado en XML.	<i>void.</i>
printTrkC	-	Imprimir el tag Track en formato basado en XML.	<i>void.</i>
printStruct	La estructura tipo data.	Imprimir la estructura, cargada previamente, en formato basado en XML.	<i>void.</i>
cargarStruct	Un puntero a char, un puntero a la estructura tipo data y una estructura tipo fecha.	Cargar la estructura tipo data con los datos procesados que se deben imprimir.	<i>bool.</i> Devuelve verdadero si la estructura cargo satisfactoriamente. Retorna falso si algún dato procesado falló.
takeArgs	El int argc, un vector de punteros a char, un puntero doble a char y un puntero a una estructura tipo fecha.	Leer los argumentos recibidos por el programa y verifica que sean los correctos.	<i>status_t.</i> Retorna ST_HELP si se reciben los argumentos -H, -h, --help. Retorna ST_INV si hubo algún error en el argumento recibido. Retorna ST_OK si los argumentos se recibieron de manera correcta.
printHelp	-	Imprimir un mensaje de ayuda.	<i>void.</i>
nmea_checksum	Un puntero a una constante <i>char</i>	Realizar el cálculo de la suma de verificación.	<i>unsigned char.</i>
checkLine	Un puntero a un <i>char</i> .	Verificar que la frase leída por el programa tenga el tipo \$GPGGA y que la suma de verificación sea la correcta.	<i>bool.</i> Devuelve verdadero si la frase interpretada es correcta. Retorna falso si la línea no cumple con los requisitos.
convertirLon	Una cadena constante y un puntero a <i>char</i>	Convertir la longitud en NMEA teniendo en cuenta si el indicador de longitud es E o S.	<i>double.</i> Retorna ERR_LATLON si la longitud interpretada no es válida. Devuelve el valor de la longitud.

convertirLat	Una cadena constante y un puntero a <i>char</i>	Convertir la latitud en NMEA teniendo en cuenta si el indicador de latitud es N o S.	<i>double</i> . Retorna ERR_LATLON si la latitud interpretada no es válida. Devuelve el valor de la latitud.
convertirCal	Un valor entero	Detectar la calidad del Fix.	<i>cal_t</i> . Retorna la calidad de Fix.
checkMembers	Dos <i>double</i> y dos valores enteros.	Verificar que los datos convertidos, la calidad del fix y los satélites utilizados sean válidos.	<i>bool</i> . Retorna falso si algunos de los datos interpretados anteriormente no son correctos.
checkDia	Un valor entero	Verificar que el día cargado sea correcto.	<i>bool</i> . Retorna falso el día cargado no se encuentra en el intervalo válido. Retorna verdadero si este es válido.
checkMes	Un valor entero	Verificar que el mes cargado sea correcto.	<i>bool</i> . Retorna falso el mes cargado no se encuentra en el intervalo válido. Retorna verdadero si este es válido.

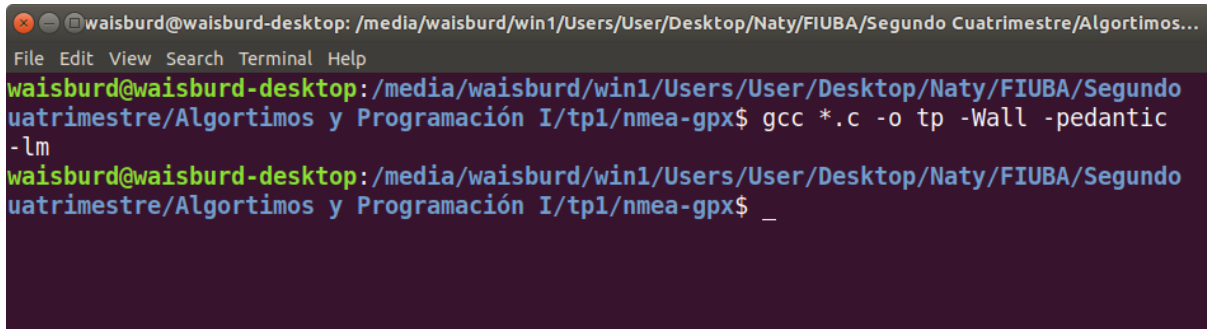
checkAnio	Un valor entero	Verificar que el año cargado sea correcto.	<i>bool</i> . Retorna falso el año cargado no se encuentra en el intervalo válido. Retorna verdadero si este es válido.
cargarFecha	Un puntero a <i>char</i> , un puntero a una estructura de tipo fecha.	Cargar la fecha y verificar si se cargo bien.	<i>bool</i> . Devuelve falso si el número ingresado no es válido o si la fecha no cumple con la cantidad de dígitos admitidos.
defaultFecha	Un puntero a una estructura de tipo fecha	Asigna la fecha del sistema a las variables de la estructura tipo fecha	<i>void</i> .
checkNum	Un puntero a <i>char</i>	Verificar que el número de la fecha recibida por la función es válida.	<i>bool</i> . Retorna falso si la fecha contiene números que no se encuentran en el intervalo del 0 al 9. Devuelve verdadero si el número es válido.

Además dividimos el programa en los archivos `tp.c` que contiene al `main`; `printFuncs.c` donde se encuentran las funciones que imprimen por `stdout`; `convFuncs.c` que contiene funciones de conversión de los datos ingresados en las sentencias `$GPGGA`; `checkFuncs.c` donde se ubican las funciones de validación de datos; y `args.c` en donde están las funciones relacionadas con los argumentos que toma la función. También creamos los headers `structData.h` (contiene declaraciones de estructuras e `#includes`), `print.h` (con los prototipos de las funciones de impresión), y `check.h` (con `#includes` y prototipos de las funciones de validación de datos).

3. Explicación de alternativas consideradas y estrategias adoptadas

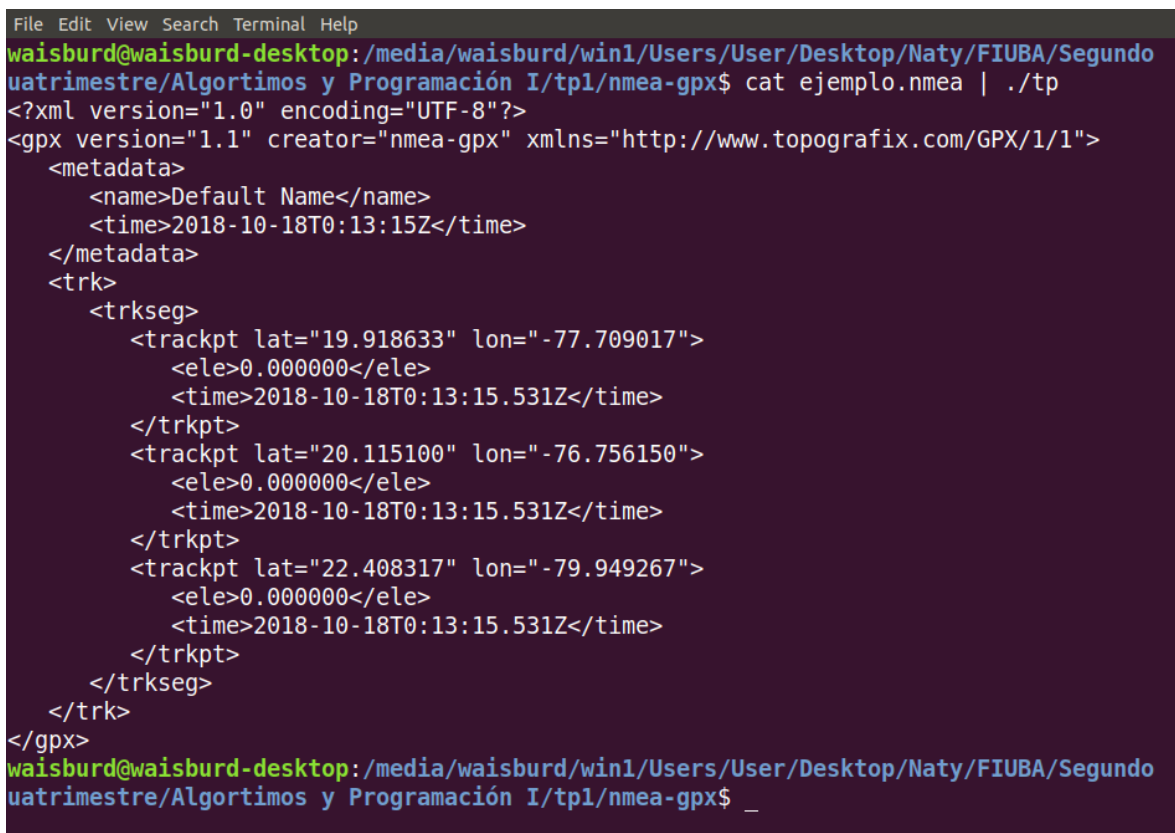
- ☐ Sólo se interpretarán las frases de tipo GGA, toda otra línea que aparezca en el flujo de entrada es ignorada silenciosamente.
- ☐ Asumimos que cada línea proveniente de `stdin` trae consigo “ * “.
- ☐ Si no nos pasan como argumento el nombre, lo tomamos como “Default Name”.
- ☐ Los parámetros que indican el tipo de argumento que se va a recibir (por ejemplo: -m para el mes, es decir un guión seguida de una letra) pueden ser ingresados como mayuscula o minuscula, es indistinto para el programa.
- ☐ Los parámetros de tipo doble guión seguido de una palabra, deben ser escritos en minúscula.
- ☐ El rango del argumento
 - ↳ día (asumimos que es entero) es de: $0 < \text{día} < 32$.
 - ↳ mes (asumimos que es entero) es de : $0 < \text{mes} < 13$.
 - ↳ año (asumimos que es entero) es de: $0 < \text{año} < 9999$.
 - Fuera de esos rangos y si no son números, devolveremos un error.
- ☐ Se tomará como fecha para los trackpts la última fecha válida ingresada por el usuario, tomando como valores default los del sistema.
- ☐ Si la fecha no es ingresada se toma la del sistema.
- ☐ Si la fecha ingresada está incompleta, se rellenará con los datos del sistema.
- ☐ Para convertir la latitud, no puede ser mayor a 90° ni menor a 0° .
- ☐ Para convertir la longitud, no puede ser mayor a 180° ni menor a 0° .
- ☐ Si los cardinales detectados no corresponden a Norte, Sur, Este o Oeste, consideramos error.
- ☐ Si el cardinal indicado es Sur, en el caso de la latitud, u Oeste, en el caso de la longitud, se multiplicará por -1 la latitud interpretada.
- ☐ La latitud no puede ser mayor a 90° ni menor a -90° .
- ☐ La longitud no puede ser menor a 180° ni menor a -180° .
- ☐ La calidad máxima del Fix es 8 y la mínima 0.
- ☐ La cantidad máxima de satélites es 12, mientras que la mínima es 0.
- ☐ El rango de números válidos para cargar la fecha es $0 \leq \text{num} \leq 9$.

4. Resultados de ejecución del programa



```
waisburd@waisburd-desktop: /media/waisburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo Cuatrimestre/Algortimos...
File Edit View Search Terminal Help
waisburd@waisburd-desktop: /media/waisburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algortimos y Programación I/tp1/nmea-gpx$ gcc *.c -o tp -Wall -pedantic
-lm
waisburd@waisburd-desktop: /media/waisburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algortimos y Programación I/tp1/nmea-gpx$ _
```

Imagen [1]: se muestra la compilación del programa utilizando el gcc. El programa no tiene ni errores ni warnings en la compilación.



```
File Edit View Search Terminal Help
waisburd@waisburd-desktop: /media/waisburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algortimos y Programación I/tp1/nmea-gpx$ cat ejemplo.nmea | ./tp
<?xml version="1.0" encoding="UTF-8"?>
<gpx version="1.1" creator="nmea-gpx" xmlns="http://www.topografix.com/GPX/1/1">
  <metadata>
    <name>Default Name</name>
    <time>2018-10-18T0:13:15Z</time>
  </metadata>
  <trk>
    <trkseg>
      <trackpt lat="19.918633" lon="-77.709017">
        <ele>0.000000</ele>
        <time>2018-10-18T0:13:15.531Z</time>
      </trackpt>
      <trackpt lat="20.115100" lon="-76.756150">
        <ele>0.000000</ele>
        <time>2018-10-18T0:13:15.531Z</time>
      </trackpt>
      <trackpt lat="22.408317" lon="-79.949267">
        <ele>0.000000</ele>
        <time>2018-10-18T0:13:15.531Z</time>
      </trackpt>
    </trkseg>
  </trk>
</gpx>
waisburd@waisburd-desktop: /media/waisburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algortimos y Programación I/tp1/nmea-gpx$ _
```

Imagen [2]: el programa se corre en modo “default” es decir, sin pasarle argumentos. Se puede ver que imprime sólo las líneas que cumplen con las condiciones determinadas por checkLine y checkMembers.


```
File Edit View Search Terminal Help
waissburd@waissburd-desktop: /media/waissburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algoritmos y Programación I/tp1/nmea-gpx$ cat ejemplo.nmea | ./tp -n nme
a
<?xml version="1.0" encoding="UTF-8"?>
<gpx version="1.1" creator="nmea-gpx" xmlns="http://www.topografix.com/GPX/1/1">
  <metadata>
    <name>nmea</name>
    <time>2018-10-18T0:20:20Z</time>
  </metadata>
  <trk>
    <trkseg>
      <trackpt lat="19.918633" lon="-77.709017">
        <ele>0.000000</ele>
        <time>2018-10-18T0:20:20.332Z</time>
      </trackpt>
      <trackpt lat="20.115100" lon="-76.756150">
        <ele>0.000000</ele>
        <time>2018-10-18T0:20:20.332Z</time>
      </trackpt>
      <trackpt lat="22.408317" lon="-79.949267">
        <ele>0.000000</ele>
        <time>2018-10-18T0:20:20.332Z</time>
      </trackpt>
    </trkseg>
  </trk>
</gpx>
waissburd@waissburd-desktop: /media/waissburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algoritmos y Programación I/tp1/nmea-gpx$ _
```

Imagen [3]: se corre el programa con el parámetro -n, y se ejecuta correctamente tomando el nombre ingresado como el name de la metadata.

```
File Edit View Search Terminal Help
waissburd@waissburd-desktop: /media/waissburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algoritmos y Programación I/tp1/nmea-gpx$ cat ejemplo.nmea | ./tp -y 199
9
<?xml version="1.0" encoding="UTF-8"?>
<gpx version="1.1" creator="nmea-gpx" xmlns="http://www.topografix.com/GPX/1/1">
  <metadata>
    <name>Default Name</name>
    <time>2018-10-18T0:27:49Z</time>
  </metadata>
  <trk>
    <trkseg>
      <trackpt lat="19.918633" lon="-77.709017">
        <ele>0.000000</ele>
        <time>1999-10-18T0:27:49.447Z</time>
      </trackpt>
      <trackpt lat="20.115100" lon="-76.756150">
        <ele>0.000000</ele>
        <time>1999-10-18T0:27:49.447Z</time>
      </trackpt>
      <trackpt lat="22.408317" lon="-79.949267">
        <ele>0.000000</ele>
        <time>1999-10-18T0:27:49.447Z</time>
      </trackpt>
    </trkseg>
  </trk>
</gpx>
waissburd@waissburd-desktop: /media/waissburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algoritmos y Programación I/tp1/nmea-gpx$ _
```

Imagen [4]: se corre el programa indicando el año de los trackpt a través de la opción -y. Vemos que funciona correctamente, y que el resto de los datos de la fecha son los últimos datos válidos cargados (en este caso la fecha del sistema).

```
File Edit View Search Terminal Help
waissburd@waissburd-desktop: /media/waissburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algortimos y Programación I/tp1/nmea-gpx$ cat ejemplo.nmea | ./tp -m 1
<?xml version="1.0" encoding="UTF-8"?>
<gpx version="1.1" creator="nmea-gpx" xmlns="http://www.topografix.com/GPX/1/1">
  <metadata>
    <name>Default Name</name>
    <time>2018-10-18T0:28:19Z</time>
  </metadata>
  <trk>
    <trkseg>
      <trackpt lat="19.918633" lon="-77.709017">
        <ele>0.000000</ele>
        <time>2018-1-18T0:28:19.640Z</time>
      </trackpt>
      <trackpt lat="20.115100" lon="-76.756150">
        <ele>0.000000</ele>
        <time>2018-1-18T0:28:19.640Z</time>
      </trackpt>
      <trackpt lat="22.408317" lon="-79.949267">
        <ele>0.000000</ele>
        <time>2018-1-18T0:28:19.640Z</time>
      </trackpt>
    </trkseg>
  </trk>
</gpx>
waissburd@waissburd-desktop: /media/waissburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algortimos y Programación I/tp1/nmea-gpx$ _
```

Imagen [5]: se corre el programa indicando el mes de los trackpt con la opción -m. Se ejecuta correctamente y modifica los datos que corresponden.

```
File Edit View Search Terminal Help
waissburd@waissburd-desktop: /media/waissburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algoritmos y Programación I/tp1/nmea-gpx$ cat ejemplo.nmea | ./tp -d 7
<?xml version="1.0" encoding="UTF-8"?>
<gpx version="1.1" creator="nmea-gpx" xmlns="http://www.topografix.com/GPX/1/1">
  <metadata>
    <name>Default Name</name>
    <time>2018-10-18T0:28:30Z</time>
  </metadata>
  <trk>
    <trkseg>
      <trackpt lat="19.918633" lon="-77.709017">
        <ele>0.000000</ele>
        <time>2018-10-7T0:28:30.888Z</time>
      </trackpt>
      <trackpt lat="20.115100" lon="-76.756150">
        <ele>0.000000</ele>
        <time>2018-10-7T0:28:30.888Z</time>
      </trackpt>
      <trackpt lat="22.408317" lon="-79.949267">
        <ele>0.000000</ele>
        <time>2018-10-7T0:28:30.888Z</time>
      </trackpt>
    </trkseg>
  </trk>
</gpx>
waissburd@waissburd-desktop: /media/waissburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algoritmos y Programación I/tp1/nmea-gpx$ _
```

Imagen [6]: se corre el programa utilizando la opción -d por lo que se cambia el dato “día” de los trackpts.

```
File Edit View Search Terminal Help
waissburd@waissburd-desktop: /media/waissburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algoritmos y Programación I/tp1/nmea-gpx$ cat ejemplo.nmea | ./tp --form
at 20201215
<?xml version="1.0" encoding="UTF-8"?>
<gpx version="1.1" creator="nmea-gpx" xmlns="http://www.topografix.com/GPX/1/1">
  <metadata>
    <name>Default Name</name>
    <time>2018-10-18T0:30:26Z</time>
  </metadata>
  <trk>
    <trkseg>
      <trackpt lat="19.918633" lon="-77.709017">
        <ele>0.000000</ele>
        <time>2020-12-15T0:30:26.585Z</time>
      </trackpt>
      <trackpt lat="20.115100" lon="-76.756150">
        <ele>0.000000</ele>
        <time>2020-12-15T0:30:26.585Z</time>
      </trackpt>
      <trackpt lat="22.408317" lon="-79.949267">
        <ele>0.000000</ele>
        <time>2020-12-15T0:30:26.585Z</time>
      </trackpt>
    </trkseg>
  </trk>
</gpx>
waissburd@waissburd-desktop: /media/waissburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algoritmos y Programación I/tp1/nmea-gpx$ _
```

Imagen [7]: corremos el programa con la opción --format y se puede ver que la fecha de los trackpts fue reemplazada por la ingresada.

```
File Edit View Search Terminal Help
waissburd@waissburd-desktop:/media/waissburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algortimos y Programación I/tp1/nmea-gpx$ cat ejemplo.nmea | ./tp --year
2030 --month 3 --day 14 --name nmea
<?xml version="1.0" encoding="UTF-8"?>
<gpx version="1.1" creator="nmea-gpx" xmlns="http://www.topografix.com/GPX/1/1">
  <metadata>
    <name>nmea</name>
    <time>2018-10-18T0:29:33Z</time>
  </metadata>
  <trk>
    <trkseg>
      <trackpt lat="19.918633" lon="-77.709017">
        <ele>0.000000</ele>
        <time>2030-3-14T0:29:33.353Z</time>
      </trackpt>
      <trackpt lat="20.115100" lon="-76.756150">
        <ele>0.000000</ele>
        <time>2030-3-14T0:29:33.353Z</time>
      </trackpt>
      <trackpt lat="22.408317" lon="-79.949267">
        <ele>0.000000</ele>
        <time>2030-3-14T0:29:33.353Z</time>
      </trackpt>
    </trkseg>
  </trk>
</gpx>
waissburd@waissburd-desktop:/media/waissburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algortimos y Programación I/tp1/nmea-gpx$ _
```

Imagen [8]: se corre el programa ingresando varios datos de fecha a través de las opciones --year, --month, --day, lo cual funciona correctamente y comprueba que los argumentos puedan ingresarse con doble guión y palabra completa.

```
File Edit View Search Terminal Help
waissburd@waissburd-desktop:/media/waissburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algortimos y Programación I/tp1/nmea-gpx$ cat ejemplo.nmea | ./tp -F 203
011151
Ingrese un dato valido
waissburd@waissburd-desktop:/media/waissburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algortimos y Programación I/tp1/nmea-gpx$ cat ejemplo.nmea | ./tp -y 100
00
Ingrese un dato valido
waissburd@waissburd-desktop:/media/waissburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algortimos y Programación I/tp1/nmea-gpx$ cat ejemplo.nmea | ./tp -m 13
Ingrese un dato valido
waissburd@waissburd-desktop:/media/waissburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algortimos y Programación I/tp1/nmea-gpx$ cat ejemplo.nmea | ./tp -d 32
Ingrese un dato valido
waissburd@waissburd-desktop:/media/waissburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algortimos y Programación I/tp1/nmea-gpx$ _
```

Imagen [9]: se corre el programa indicando opciones e ingresando datos inválidos, por lo que la función retorna 0 y le indica al usuario que debe ingresar un dato válido.

```
File Edit View Search Terminal Help
waisburd@waisburd-desktop:/media/waisburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algoritmos y Programación I/tp1/nmea-gpx$ cat ejemplo.nmea | ./tp --year
1997 -f 20201215
<?xml version="1.0" encoding="UTF-8"?>
<gpx version="1.1" creator="nmea-gpx" xmlns="http://www.topografix.com/GPX/1/1">
  <metadata>
    <name>Default Name</name>
    <time>2018-10-18T0:31:29Z</time>
  </metadata>
  <trk>
    <trkseg>
      <trackpt lat="19.918633" lon="-77.709017">
        <ele>0.000000</ele>
        <time>2020-12-15T0:31:29.178Z</time>
      </trackpt>
      <trackpt lat="20.115100" lon="-76.756150">
        <ele>0.000000</ele>
        <time>2020-12-15T0:31:29.178Z</time>
      </trackpt>
      <trackpt lat="22.408317" lon="-79.949267">
        <ele>0.000000</ele>
        <time>2020-12-15T0:31:29.178Z</time>
      </trackpt>
    </trkseg>
  </trk>
</gpx>
waisburd@waisburd-desktop:/media/waisburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algoritmos y Programación I/tp1/nmea-gpx$ _
```

```
File Edit View Search Terminal Help
waisburd@waisburd-desktop:/media/waisburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algoritmos y Programación I/tp1/nmea-gpx$ cat ejemplo.nmea | ./tp --form
at 20201215 -Y 1998
<?xml version="1.0" encoding="UTF-8"?>
<gpx version="1.1" creator="nmea-gpx" xmlns="http://www.topografix.com/GPX/1/1">
  <metadata>
    <name>Default Name</name>
    <time>2018-10-18T0:30:49Z</time>
  </metadata>
  <trk>
    <trkseg>
      <trackpt lat="19.918633" lon="-77.709017">
        <ele>0.000000</ele>
        <time>1998-12-15T0:30:49.442Z</time>
      </trackpt>
      <trackpt lat="20.115100" lon="-76.756150">
        <ele>0.000000</ele>
        <time>1998-12-15T0:30:49.442Z</time>
      </trackpt>
      <trackpt lat="22.408317" lon="-79.949267">
        <ele>0.000000</ele>
        <time>1998-12-15T0:30:49.442Z</time>
      </trackpt>
    </trkseg>
  </trk>
</gpx>
waisburd@waisburd-desktop:/media/waisburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algoritmos y Programación I/tp1/nmea-gpx$ _
```

Imágenes [10] y [11]: se corre el programa con ambas opciones para el ingreso de fecha por línea de comandos, vemos que el programa toma como fecha para los trackpts el último dato válido ingresado.


```
File Edit View Search Terminal Help
waisburd@waisburd-desktop: /media/waisburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algoritmos y Programación I/tp1/nmea-gpx$ cat ejemplo.nmea | ./tp -N nme
a -F 20301115
<?xml version="1.0" encoding="UTF-8"?>
<gpx version="1.1" creator="nmea-gpx" xmlns="http://www.topografix.com/GPX/1/1">
  <metadata>
    <name>nmea</name>
    <time>2018-10-18T0:22:58Z</time>
  </metadata>
  <trk>
    <trkseg>
      <trackpt lat="19.918633" lon="-77.709017">
        <ele>0.000000</ele>
        <time>2030-11-15T0:22:58.012Z</time>
      </trackpt>
      <trackpt lat="20.115100" lon="-76.756150">
        <ele>0.000000</ele>
        <time>2030-11-15T0:22:58.012Z</time>
      </trackpt>
      <trackpt lat="22.408317" lon="-79.949267">
        <ele>0.000000</ele>
        <time>2030-11-15T0:22:58.012Z</time>
      </trackpt>
    </trkseg>
  </trk>
</gpx>
waisburd@waisburd-desktop: /media/waisburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algoritmos y Programación I/tp1/nmea-gpx$ _
```

Imagen [12]: corremos el programa indicando las opciones en mayúscula, todo se ejecuta normalmente.

```
File Edit View Search Terminal Help
waisburd@waisburd-desktop: /media/waisburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algoritmos y Programación I/tp1/nmea-gpx$ cat ejemplo.nmea | ./tp --help
Utilice los argumentos:
-n o --name para ingresar el nombre de la ruta
-f o --format aaaammdd para ingresar el año, mes y día
-y o --year para ingresar el año con la centuria
-m o --month para ingresar el número de mes y
-d o --day para ingresar el número de día del mes

waisburd@waisburd-desktop: /media/waisburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algoritmos y Programación I/tp1/nmea-gpx$ cat ejemplo.nmea | ./tp -h
Utilice los argumentos:
-n o --name para ingresar el nombre de la ruta
-f o --format aaaammdd para ingresar el año, mes y día
-y o --year para ingresar el año con la centuria
-m o --month para ingresar el número de mes y
-d o --day para ingresar el número de día del mes

waisburd@waisburd-desktop: /media/waisburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algoritmos y Programación I/tp1/nmea-gpx$ cat ejemplo.nmea | ./tp -H
Utilice los argumentos:
-n o --name para ingresar el nombre de la ruta
-f o --format aaaammdd para ingresar el año, mes y día
-y o --year para ingresar el año con la centuria
-m o --month para ingresar el número de mes y
-d o --day para ingresar el número de día del mes

waisburd@waisburd-desktop: /media/waisburd/win1/Users/User/Desktop/Naty/FIUBA/Segundo
uatrimestre/Algoritmos y Programación I/tp1/nmea-gpx$ _
```

Imagen [13]: corremos el programa varias veces utilizando las opciones --help, -h y -H respectivamente, en todos los casos se imprime una ayuda acerca de las opciones y su función.

5. Archivos de prueba utilizados:

ejemplo.nma (Fue dado por el profesor.)

gps.txt (Hecho por los desarrolladores del programa.)

6. Reseña de problemas encontrados en el desarrollo del programa y soluciones:

En primer lugar, en vez de la función `cargarStruct`, habíamos implementado una función que retornaba la estructura cargada. Esto no nos permitía validar que la misma se haya cargado correctamente, con todos sus datos válidos. Es por eso que la reemplazamos por la función `cargarStruct` que retorna un bool dependiendo de si pudo cargar la estructura, para lo cual todos los datos deben ser válidos (comprobado por `checkMembers`), o false si no pudo realizarlo, en cuyo caso se descarta la línea silenciosamente.

Otro problema que encontramos fue en el archivo `args.c` en la función `takeArgs`. En este caso al compilar recibíamos en la consola “Segmentation fault (core dumped)”, esto sucedía porque en caso de que ingresaran mal un argumento, el programa intentaba acceder a una dirección de memoria a la cual no se tenía acceso. Esto lo solucionamos agregando la validación con `else if(argc > i + 1)`, que primero se fija si el lugar de memoria al que estamos tratando de acceder (`argv[i+1]`) forma parte de `argv[]`.

Finalmente nos encontramos con el problema de que la fecha para los `trackpts` lleva los segundos como float y no como int, hasta los milisegundos. Para poder obtener este número, implementamos la función `gettimeofday` incluida en la librería `<sys/time.h>`. Esto modifica una estructura de tipo `timeval`, cuyo miembro `tv_usec` posee la cantidad de microsegundos. Por lo tanto, a nuestra estructura de tipo fecha, en el miembro segundos le asignamos el valor de segundos del sistema más la cantidad de microsegundos dividida por

1 millón para pasarla a segundos. Como se muestran hasta los milisegundos, al imprimir este valor se indica en la función print “%.3f”.

7. Indicaciones sobre la compilación:

Se debe compilar todos los archivos .c juntos, incluyendo el parámetro -lm al momento de hacerlo para que el compilador enlace las librerías necesarias.

8. Bibliografía:

[1] B.W. Kernighan y D.M. Ritchie. The C Programming Language. 2. a ed. Prentice-Hall software series. Prentice Hall, 1988. ISBN: 9780131103627.

[2] P. Deitel y H. Deitel. C How to Program. 7. a ed. Pearson Education, 2012. ISBN: 9780133061567.

[3] <http://docs.mis-algoritmos.com/c.funcion.atol.html> : para la funcion atol.

[4] https://www.tutorialspoint.com/c_standard_library/math_h.htm

[5] <http://man7.org/linux/man-pages/man2/gettimeofday.2.html>

[6] https://www.tutorialspoint.com/c_standard_library/stdlib_h.htm

[7] https://www.tutorialspoint.com/c_standard_library/string_h.htm

[8] https://www.tutorialspoint.com/c_standard_library/time_h.htm

[9] <http://pubs.opengroup.org/onlinepubs/7908799/xsh/systime.h.html>