1) Segmentation fault: le programme essaie d'accéder à un espace mémoire auquel il n'est pas autorisé.

2) Le signal reçu est de type SIGSEGV. Pour vérifier cela, après l'éxecution nous tapons dans la console echo $? qui nous renvoie 139 - 128 = 11. Qui correspond au signal reçu pour terminer le processus. Puis à l'aide de la commande kill -l on voit que le signal 11 correspond à un SIGSEGV.

3) Via la pile d'appel on voit qu'une erreur est générée lors de l'appel à la fonction knot_to_kmh_str()., précisément on retourne NULL si le GPS n'est pas défini. Remarque: lors de la compilation, un warning est retournée concernant cette ligne:

```
     nmea.c:23:9: warning: null argument where non-null required
(argument 1) [-Wnonnull]
         puts(NULL);
         ^
```

4) En analysant le message retourné :

```
     libptmx.so: cannot open shared object file: No such file or
directory
```

Il sa'avère que libptmx.so n'existe pas.

5) ldd permet d'afficher les dépendances au niveau des librairies. Lors de cet appel, on constate qu'ils manquent libptmx.so et libnmea.so.

6) on doit utiliser la commande : export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$PWD/lib (run.sh) pour exporter les libraries manquantes qui se trouvent dans le dossier ../gps/lib.

7) lors du debugging 's' s'arrête à chaque logne et va entrer dans les fonctions qui sont appelés pour faire du pas à pas dans ces fonctions-ci. Tandis que 'n' ne fait pas de pas à pas dans ces fonctions.

8) Cet outil est intéressant pour les sytèmes embarqués sans utiliser GDB directement.

Exercice 2:

4) man 3 printf pour accéder à printf de la libraire et non pas de cmd
int printf(const char *format, ...);
c'est une fonction qui peut prendre un nombre de paramètres variable