单点定位（SPP）和相对定位（RTK）流程中的读取和传递过程进行如下：

总体流程概述

1. **程序入口（rtknavi.c)**：
    - main() 函数是程序的起点。它负责定义配置（如处理模式、输入/输出文件路径）并调用 rtksvrstart() 启动 RTK/GNSS 处理服务。
    - 在此文件中，paths 变量定义了数据来源，例如 "data\\Rover_20240520_082407.rtcm3"（流动站数据）和 "data\\Base_Station_20240520_082407.rtcm3"（基站数据）。
2. **数据读取与解码（rtcm.c, rtcm3.c, rcvraw.c)**：
    - 服务启动后，RTK 服务器（rtksvr）会从指定的数据流中读取原始字节数据。
    - 如果数据是 RTCM 格式，rtcm.c 中的 input_rtcm3() 会被调用来处理字节流，它负责帧同步和校验。
    - input_rtcm3() 接着调用 rtcm3.c 中的 decode_rtcm3()，该函数根据消息类型（如 1004, 1019, 1077, 1005 等）分发到具体的解码函数。
    - 如果数据是原始二进制格式（RAW），rcvraw.c 中的 decode_... 函数（如 decode_frame, decode_glostr）会负责解码。
3. **定位解算（pntpos.c)**：
    - pntpos.c 文件中的 pntpos() 函数执行**单点定位（SPP)**。它接收解码后的观测数据和导航电文作为输入，计算接收机的位置、速度和钟差。
    - 相对定位（RTK）则会使用一个（未在您提供的文件中完全展示的）rtkpos 函数，它需要同时接收流动站和基站的观测数据。

各类数据的读取与传递详情

1. 单点定位用的伪距和载波相位

读取函数和变量：

- **函数**：在 rtcm3.c 文件中，伪距和载波相位主要由 **MSM（Multiple Signal Messages）** 消息的解码函数读取，例如：
    - decode_msm4() (MT 1074, 1084, 1094, ...)
    - decode_msm5() (MT 1075, 1085, 1095, ...)

    - decode_msm6() (MT 1076, 1086, 1096, ...)
    - decode_msm7() (MT 1077, 1087, 1097, ...)
    - （也包括旧版的 decode_type1002 和 decode_type1004 等）
- **读取变量**：这些函数从 rtcm->buff（原始字节缓冲区）中解码数据。
    - 在 decode_msm4 和 decode_msm6 中，伪距信息被读入局部变量 prv，载波相位信息被读入 cpv。
    - 在 decode_msm5 和 decode_msm7 中，它们同样被读入 prv 和 cpv。
    - 在 decode_type1002 (L1) 中，伪距读入 pr1，载波相位相关值读入 ppr1。
    - 在 decode_type1004 (L1/L2) 中，伪距读入 pr1 和 pr21，载波相位相关值读入 ppr1 和 ppr2。

传递变量：

1. 解码函数（如 decode_msm4）会调用 save_msm_obs() 函数。
2. save_msm_obs() 将伪距和载波相位（转换成米和周）存储到 rtcm_t 结构体中的观测数据数组里：
    - rtcm->obs.data[index].P (伪距)
    - rtcm->obs.data[index].L (载波相位)
3. 这个 rtcm->obs 结构（类型为 obs_t）随后被传递给 pntpos.c 中的 pntpos() 函数，作为参数 obs。
4. 在 pntpos() 内部，obs 被传递给 estpos()，后者再调用 rescode()。
5. 在 rescode() 中，调用 prange() 函数，**最终使用 obs->P（伪距）来计算单点定位**。
    - **注意**：标准的单点定位（SPP）仅使用伪距（obs->P）来计算位置（在 estpos 中）。载波相位（obs->L）虽然在同一结构体中被读取和传递，但 SPP 流程不使用它来定位。SPP 的速度估算（在 estvel 中）使用的是多普勒观测值（obs->D）。

2. 单点定位用的导航电文（星历）

读取函数和变量：

- **函数**：导航电文的读取同样发生在 rtcm3.c 和 rcvraw.c 中。

    - rtcm3.c：
        - decode_type1019()：解码 GPS 广播星历。
        - decode_type1020()：解码 GLONASS 广播星历。
        - decode_type1042() / decode_type63()：解码 BDS 广播星历。
        - decode_type1044()：解码 QZSS 广播星历。
        - decode_type1045() / decode_type1046()：解码 Galileo 广播星历。
    - rcvraw.c：
        - decode_frame()：解码 GPS/QZSS 的原始导航电文子帧。
        - decode_glostr()：解码 GLONASS 的原始导航电文字符串。
        - （以及 decode_bds_d1, decode_gal_inav 等其他系统的原始解码函数）
    - preceph.c：
        - readsp3()：读取 SP3 格式的精密星历文件（用于精密单点定位，而非标准单点定位）。
- **读取变量**：这些函数从 rtcm->buff 或 raw->buff（原始字节缓冲区）中提取电文参数（如开普勒根数、钟差参数等），并将它们临时存放在一个局部的 eph_t（GPS/GAL/BDS 等）或 geph_t（GLONASS）结构体变量中。

传递变量：

1. 解码函数（如 decode_type1019）将解码后的 eph_t 结构体赋值给 rtcm->nav.eph[sat-1]。
2. decode_type1020 将 geph_t 结构体赋值给 rtcm->nav.geph[prn-1]。
3. 所有这些星历数据被汇总到 rtcm->nav（或 raw->nav）结构体中，该结构体的类型是 nav_t。
4. nav_t 结构体（作为 nav 参数）被传递给 pntpos.c 中的 pntpos() 函数。
5. pntpos() 接着将 nav 传递给 ephemeris.c: satposs()。
6. satposs() 再将 nav 传递给 ephemeris.c: satpos()。
7. satpos() 最终将 nav 传递给 ephpos()，ephpos() 内部调用 seleph() 或 selgeph() 来从 nav->eph 和 nav->geph 数组中搜索并选择最合适的星历用

于计算。

3. 相对定位用的数据

相对定位（RTK）需要来自**基准站**和**流动站**两组的观测数据，以及基准站的精确坐标。

读取函数和变量：

- **函数**：
    - rtknavi.c: main()：通过 paths 变量定义了基站和流动站的数据源（例如 ...Base_Station...rtcm3 和 ...Rover...rtcm3）。
    - rtcm.c: input_rtcm3() 和 rtcm3.c: decode_rtcm3()：如上所述，这是读取和解码 RTCM3 数据的核心入口。
    - **观测数据**：使用 decode_msm4 到 decode_msm7 系列函数（MT 1074-1077, 1084-1087 等）来读取两个站点的伪距、载波相位和信噪比。
    - **基站坐标**：使用 decode_type1005 或 decode_type1006 来读取基准站的天线参考点（ARP）坐标。
- **读取变量**：
    - **观测数据**：解码后的观测值（伪距、载波相位等）被读入 rtcm->obs.data 数组。
    - **基站坐标**：在 decode_type1005 或 decode_type1006 中，坐标被读入局部变量 rr，然后存储在 rtcm->sta.pos 中。

传递变量：

- 在 rtknavi 启动的 RTK 服务中（rtksvr.c，未提供），服务器会为**每个数据流**（即流动站一个，基站一个）维护一个 rtcm_t 结构体。
- 当 rtcm.c / rtcm3.c 解码数据时：
    1. 来自**流动站**流的数据（例如 paths[0]） 被解码并存入 rtksvr.obs[0]（流动站观测数据）和 rtksvr.nav（导航电文）。
    2. 来自**基站**流的数据（例如 paths[1]） 被解码并存入 rtksvr.obs[1]（基站观测数据）和 rtksvr.sta[1]（基站坐标）。
- 随后，RTK 处理函数（例如 rtkpos，在您提供的文件中未包含，但 lambda.c 是为其服务的）会被调用，并同时接收**流动站观测数据**（如 rtksvr.obs[0]）、**基站观测数据**（如 rtksvr.obs[1]）、**导航电文**（rtksvr.nav）和**基站坐标**（rtksvr.sta[1]）作为参数，以计算高精度的相对定位解。