**Wycieki pamięci**

**Narzędzia**

Do znalezienia wycieków pamięci używaliśmy profilera pamięci oraz debbugera wbudowanego w Microsoft Visual Studio 2019 16.11.5.

**Metody**

W celu znalezienia wycieku ustawialiśmy breakpoint na początku i na końcu programu. Następnie kompilowaliśmy i uruchamialiśmy program przy pomocy debbugera. Kiedy program zatrzymywał się na breakpoincie tworzyliśmy snapshot pamięci. Na koniec wystarczyło porównać pamięć zajmowaną przez program na początku i na końcu jego działania.

**Początkowy stan programu**

W wersji pierwotnej program zawierał następujące wycieki pamięci:

1. Przy użyciu implementacji make\_spl() z pliku interpolator.c A screenshot of a computer

   Description automatically generated with medium confidence
2. Przy użyciu implementacji make\_spl() z pliku prosta.c A screenshot of a computer

   Description automatically generated with medium confidence
3. Przy użyciu implementacji make\_spl() z pliku aproksymator\_na\_bazie.c Graphical user interface

   Description automatically generated

Jak można zobaczyć na powyższych zrzutach ekranu część wycieków jest wspólna dla wszystkich metod i ma swoje źródło bezpośrednio w funkcji main().

**Wycieki w funkcji main()**

Wyciek wynika z niezwalniania pamięci alokowanej przez funkcję read\_pts\_failed(). Rozwiązaniem jest zwolnienie tej pamięci we wszystkich ścieżkach wykonywania programu.

**Wycieki w interpolator.c**

Pierwszy wyciek pamięci jest związany z wywołaniem funkcji alloc\_spl(). W celu zwalczenia go zwalniamy pamięć zajmowaną przez zmienną line w funkcji main().

Drugim wyciekiem pamięci jest przy alokowaniu macierzy. Aby rozwiązać problem zwalniamy zajmowaną przez macierz pamięć na końcu funkcji make\_spl().

**Pozostałe wycieki**

Wycieki pamięci w dwóch pozostałych plikach są albo rozwiązywane przez naprawę pierwszego wycieku w pliku interpolator.c albo ich naprawa jest analogiczna do drugiego wycieku.

Na powyższych zrzutach ekranu można zaobserwować również alokacje w main() oraz \_vfprintf\_l(). Alokacja w main dotyczy otwierania pliku i nie powinna być zwalniana w sposób inny niż poprzez wywołanie fclose(), co jest już robione, dlatego można to zignorować. Druga funkcja nie należy do naszego kodu, dlatego również można ją zignorować.