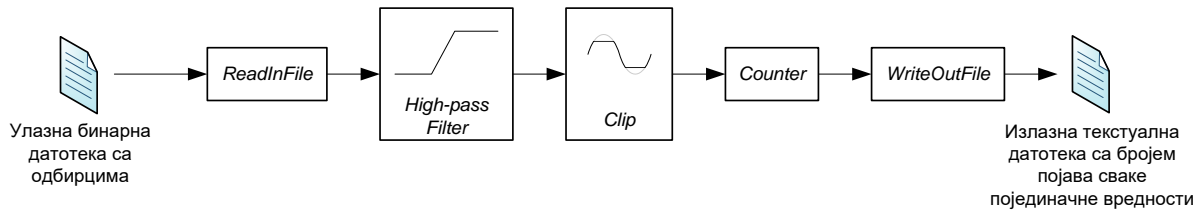


УВОЂЕЊЕ ПАРАЛЕЛИЗМА КОРИШЋЕЊЕМ TBV БИБЛИОТЕКЕ – СЛОЖЕНЕ ПЕТЉЕ И КОНТЕЈНЕРИ

Блокови обраде сигнала

Ову вежбу прате два пројекта: *Serial* и *Parallel*. У пројекту *Serial* реализована је секвенцијална имплементација блокова обраде сигнала са слике. Обрада се састоји од филтрирања одбирака из улазне датотеке *HIGH-PASS* филтром, одсецања вредности већих од задате и пребројавање појава сваке појединачне вредности.



Слика 1: Пример блокова обраде сигнала

Детаљи о појединачним блоковима:

- *ReadInFile* – читање одбирака из улазне датотеке,
- *HighPassFilter* – филтрирање прочитаних одбирака високопропусним филтром; филтрирање се изводи по формули
$$y[i] := \alpha * (y[i-1] + x[i] - x[i-1]),$$
- *Clip* – ограничавање вредности филтрираних одбирака на задату вредност,
- *Counter* – бројање појава сваке појединачне вредности (*histogram*) и
- *WriteOutFile* – упис броја појава појединачних вредности у излазну датотеку.

Пројекат *Parallel* је подешен да укључује TBV библиотеку и садржи секвенцијалну имплементацију блокова обраде као и пројекат *Serial*. У њему је потребно реализовати паралелне имплементације описаних блокова обраде, према упутствима из задатка који су дата у даљем тексту. За проверу исправности рада паралелне верзије програма користити излазну датотеку.

Урадити паралелизацију филтрирања из серијског кода коришћењем *tbb*-а:

1. Изменити модул/функцију *Clip* тако да свој излаз уписује у конкурентни вектор (*tbb::concurrent_vector*).

2. Бројање вредности у функцији Counter реализовати коришћењем конкурентне мапе (*tbb::concurrent_hash_map*). Користити структуру *parallel_for*. Прилагодити модул/функцију WriteOutFile тако да користи конкурентну мапу.
3. Изменити модул/функцију HighPassFilter тако да свој излаз уписује у конкурентни вектор.
4. Изменити модул/функцију Clip да користи структуру *parallel_for*.

НАПОМЕНА: Након сваког завршеног задатка проверити исправност програма.