

Област вежби: Паралелно програмирање

УВОЂЕЊЕ ПАРАЛЕЛИЗМА КОРИШЋЕЊЕМ ТВВ БИБЛИОТЕКЕ – ПРОТОЧНА СТРУКТУРА

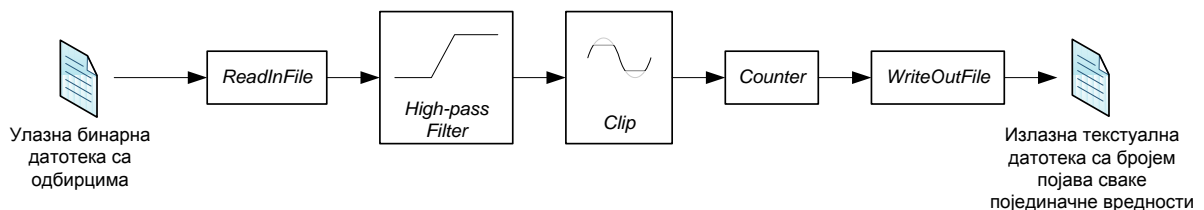
Предуслови:

- Rpi2 рачунар (без додатака),
- Преводиоц *GCC* освежен на верзију 4.7 или новију,
- Преузета и подешена ТВВ библиотека на Raspberry Pi уређају, према опису из документа „УВОД - Raspberry Pi рачунар“,
- Подешен мрежни приступ на један од начина представљених у документу „УВОД - Raspberry Pi рачунар“ уколико се ради преко мреже. Ако се Rpi2 рачунар користи као самосталан рачунар овај захтев се може занемарити,
- Познавање језика Це и материјала из вежби „ТВВ, ТУТОРИЈАЛ I - СЛОЖЕНЕ ПЕТЉЕ И КОНТЕЈНЕРИ“ и „ТВВ, ТУТОРИЈАЛ I I - СЛОЖЕНЕ ПЕТЉЕ – ПРОТОЧНА ОБРАДА“.

Увод

Блокови обраде сигнала

Ову вежбу прате два пројекта. У пројекту *serial* се налази 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 *pipeline* је празан. Пројекат је подешен да укључује TBV библиотеку. У њему је потребно реализовати паралелне имплементације описаних блокова обраде, према упутствима из задатка у даљем тексту. Најбоље је поћи од серијске имплементације, копирањем датотека из тог пројекта у овај пројекат.

Задатак

Блокове филтера реализовати као проточну структуру у пројекту *pipeline*, који је приложен уз ову вежбу:

- 1) Све контејнере заменити одговарајућим баферима. Бафер је потребно направити, најбоље као *template* класу/структуру јер ће бити потребан за различите типове. Можете користити приложену имплементацију “*PipelineBuffer*”.
- 2) Читање улазне датотеке, *high-pass* филтар, одсецање и бројање вредности, прерадити тако да буду засебне фазе у проточној обради (неке од њих могу бити паралелне, неке морају бити серијске) и
- 3) Унутар проточних фаза користити паралелне структуре (*parallel_for*), где је то могуће.