# Readme

*Running the RWP*

* The project was executed on the **Simple-scaler using the SPEC2000 integer and floating point benchmarks.**
* To execute following files were edited/hacked:

**cache.c and cache.h :-** As mentioned in the Appendix of the report the changes are made to accommodate the read-write partitioning.

* We have added an additional switch to run the ***read-write partitioning (RWP), w*** analogous to l for LRU.
* The command that were used for the running the ***read-write partitioning*** *is:*

./RUN$1 ../../simplesim-3.0/sim-outorder ../../spec2000binaries/$1\*.peak.ev6 -max:inst 50000000 -fastfwd 20000000 -redir:sim $1\_sim\_output\_rwp\_32.log -bpred bimod -bpred:bimod 256 -bpred:ras 8 -bpred:btb 64 2 -cache:dl1 dl1:128:32:4:w -cache:dl2 ul2:1024:64:4:w

where,

w is for ***read-write partitioning***

* **Running the RWP:** To run the RWP with standard Simple-scaler, two files cache.c and cache.h needs to be replaced with our provided version of files and need to be re-compiled with alpha version. Then the use of the above command will execute the RWP for the selected benchmark ($1).
* The data that was collected for the report and analysis is placed in a single spread-sheet named “RWP.xls” and will be a part of the compressed deliverable.
* To get the trace of the memory operation we used simple scaler setup from the github, mentioned in the references for getting the information of load/store operation in the benchmarks.
* Teamwork

|  |  |  |  |
| --- | --- | --- | --- |
| Phase | Sub-phase | Plan | Execution |
| PHASE 1 |  | Understand the reference papers and come up with a concise plan of tasks for the project. Listing down the tasks with the  requirements will clearly help us move proactively, towards completion of the project. | Biren and Shanmati |
| PHASE 2 |  | Programming and Implementation of three different modules on the existing cache system of the simulator in sub phases. | Biren and Shanmati |
| Sub-phase1 | Implementation of partitioning the cache into clean and dirty lines | Biren and Shanmati |
| Sub-phase2 | Implementation of the algorithm to predict the partition size that maximizes read hits | Biren and Shanmati |
| Sub-phase3 | Implementation of the algorithm to predict the partition size that maximizes read hits | Biren and Shanmati |
| PHASE 3 |  | Review of the implementation with Professor and TA. Modify the implementation based on the review points. Extensive testing to gather the data needed for the term paper. | Biren and Shanmati |
| PHASE 4 |  | Project report which will conclude the findings from the gathered data and simulation results | Biren and Shanmati |