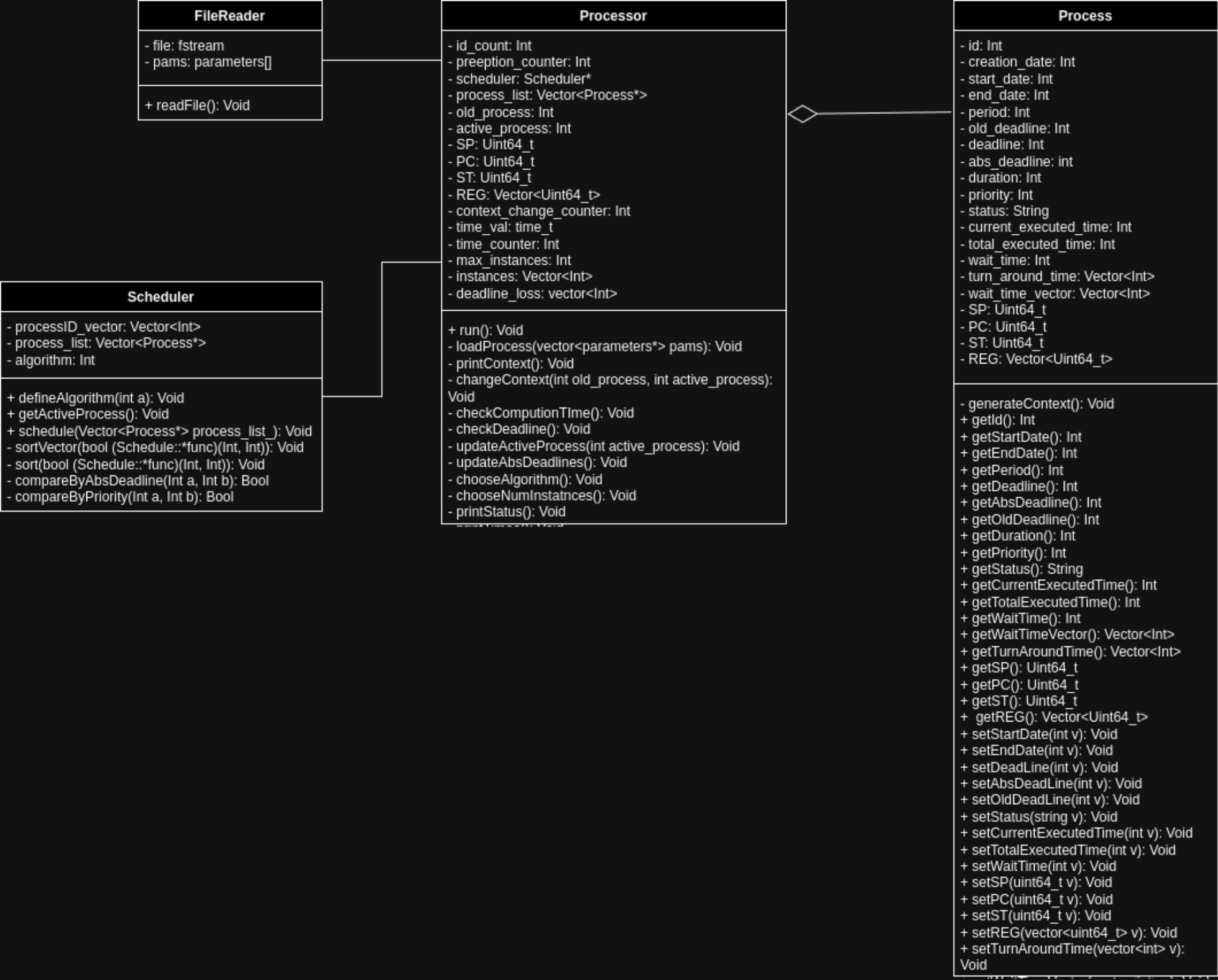
FileReader
- file: fstream
- pams: parameters[]
+ readFile(): Void
Processor
- id_count: Int
- preeption_counter: Int
- scheduler: Scheduler*
- process_list: Vector<Process*>
- old_process: Int
- active_process: Int
- SP: Uint64_t
- PC: Uint64_t
- ST: Uint64_t
- REG: Vector<Uint64_t>
- context_change_counter: Int
- time_val: time_t
- time_counter: Int
- max_instances: Int
- instances: Vector<Int>
- deadline_loss: vector<Int>
+ run(): Void
- loadProcess(vector<parameters*> pams): Void
- printContext(): Void
- changeContext(int old_process, int active_process): Void
- checkComputionTIme(): Void
- checkDeadline(): Void
- updateActiveProcess(int active_process): Void
- updateAbsDeadlines(): Void
- chooseAlgorithm(): Void
- chooseNumInstatnces(): Void
- printStatus(): Void
Process
- id: Int
- creation_date: Int
- start_date: Int
- end_date: Int
- period: Int
- old_deadline: Int
- deadline: Int
- abs_deadline: int
- duration: Int
- priority: Int
- status: String
- current_executed_time: Int
- total_executed_time: Int
- wait_time: Int
- turn_around_time: Vector<Int>
- wait_time_vector: Vector<Int>
- SP: Uint64_t
- PC: Uint64_t
- ST: Uint64_t
- REG: Vector<Uint64_t>
- generateContext(): Void
+ getId(): Int
+ getStartDate(): Int
+ getEndDate(): Int
+ getPeriod(): Int
+ getDeadline(): Int
+ getAbsDeadline(): Int
+ getOldDeadline(): Int
+ getDuration(): Int
+ getPriority(): Int
+ getStatus(): String
+ getCurrentExecutedTime(): Int
+ getTotalExecutedTime(): Int
+ getWaitTime(): Int
+ getWaitTimeVector(): Vector<Int>
+ getTurnAroundTime(): Vector<Int>
+ getSP(): Uint64_t
+ getPC(): Uint64_t
+ getST(): Uint64_t
+ getREG(): Vector<Uint64_t>
+ setStartDate(int v): Void
+ setEndDate(int v): Void
+ setDeadLine(int v): Void
+ setAbsDeadLine(int v): Void
+ setOldDeadLine(int v): Void
+ setStatus(string v): Void
+ setCurrentExecutedTime(int v): Void
+ setTotalExecutedTime(int v): Void
+ setWaitTime(int v): Void
+ setSP(uint64_t v): Void
+ setPC(uint64_t v): Void
+ setST(uint64_t v): Void
+ setREG(vector<uint64_t> v): Void
+ setTurnAroundTime(vector<int> v): Void
Scheduler
- processID_vector: Vector<Int>
- process_list: Vector<Process*>
- algorithm: Int
+ defineAlgorithm(int a): Void
+ getActiveProcess(): Void
+ schedule(Vector<Process*> process_list_): Void
- sortVector(bool (Schedule::*func)(Int, Int)): Void
- sort(bool (Schedule::*func)(Int, Int)): Void
- compareByAbsDeadline(Int a, Int b): Bool
- compareByPriority(Int a, Int b): Bool