

AULA 8 – TIPOS ABSTRATOS DE DADOS

***** Entregue, num ficheiro ZIP, os 3 ficheiros .cpp deste guião *****

Pretende-se desenvolver o tipo abstrato de dados **SCHEDULINGSEQUENCE**, que permitirá representar uma sequência de intervalos de tempo não sobreponíveis, constituindo um conjunto ordenado de tarefas que definem uma sequência de escalonamento.

Usando como base os tipos de dados desenvolvidos no guião anterior, é possível definir, numa aproximação “bottom-up”, os seguintes tipos:

- **DATETIME**, agregando, para um determinado dia, uma data e uma hora.
- **TIMEINTERVAL**, definindo um intervalo de tempo, aberto à direita.
- **SCHEDULINGSEQUENCE**, armazenando uma sequência ordenada (escalonamento) de intervalos de tempo, cada um deles associado a uma tarefa determinada.

O tipo **SCHEDULINGSEQUENCE** usa um **vetor (“array”)** para armazenar a sequência ordenada de intervalos de tempo.

TAREFAS

- Comece por analisar o conteúdo dos ficheiros disponibilizados e as funcionalidades definidas para cada tipo de dados.
- **Nota:** os ficheiros surgem com a extensão .cpp (e não .c) por compatibilidade com o **CodeCheck**; mas é usada a linguagem C.
- Compare a sua solução do guião anterior para os tipos **DATE** e **TIME** com as versões finais fornecidas.
- Conclua o desenvolvimento do tipo **DATETIME** parcialmente fornecido.
- Conclua o desenvolvimento do tipo **TIMEINTERVAL** parcialmente fornecido.
- Desenvolva e teste o tipo **SCHEDULINGSEQUENCE**.
- Teste de modo incremental todo o código desenvolvido: **CodeCheck** ou **Makefile**.
- Pode executar o **valgrind** para verificar se tem “*memory leaks*” ou outros problemas relacionados com a alocação dinâmica de memória.