

# Atividade 5 - Org. Arg. 1

Profa. Dra. Cíntia Borges Margi  
(cintia@usp.br)

Guilherme S. Salustiano  
(salustiano@usp.br)

## Contexto

Cache é um conceito universal da computação usado em diversos contextos para melhorar o desempenho de programas. Nessa atividade iremos implementar e analisar o cache para um banco de dados chave valor inspirado no cache estudado do RISC-V.

O banco de dados é o mais simples possível com apenas duas funções:

```
db_data_t get(db_key_t key);  
void put(db_key_t key, db_data_t data);
```

O put salva o valor data em um arquivo com o nome key e o get lê o arquivo com o nome key e retorna o valor salvo. Isso é propositalmente lento.

Sendo a chave um número inteiro sem sinal de 64 bits e o valor uma struct de 32 bytes definidos em database\_t.h.

## Tarefa

Você deve implementar um cache associativo com o número de conjuntos configurável pela variável ASSOCIATIVITY no arquivo database.h. Você deve usar a variável static cache\_entry\_t cache[CACHE\_SIZE]; para armazenar o cache, não sendo permitido usar outras variáveis globais. Em caso de hit você deve também incrementar o valor de cache\_hits.

Você pode baixar o código base no repositório do experimento, disponível da pasta src. Um makefile também foi fornecido para facilitar a compilação e execução do programa, para executar basta rodar make run.

Após terminar a implementação, você deve rodar o programa para diferentes valores de ASSOCIATIVITY e gerar um arquivo benchmark.txt com os resultados. Para fazer isso basta executar o comando abaixo:

```
make -s run_benchmark
```

Atenção: isso altera o valor de ASSOCIATIVITY no arquivo database.h! Para voltar ao valor padrão você pode rodar make reset.

## Entrega final

Ao final, gere um zip atv5.zip com os arquivos.

```
atvr5.zip  
├─ database.c  
└─ benchmark.txt
```