

## PAPI - Performance Application Programming Interface (*versão 4.1.3.0*)

Web: <http://icl.cs.utk.edu/projects/papi/downloads/papi-4.1.3.tar.gz>

É um interface (API) para acesso aos **contadores** de desempenho do *hardware*, que estão disponíveis nos processadores atuais.

Qualquer um dos mais de 100 **eventos** pré-definidos pode ser contado, quer através de uma interface de programação de alto nível ou de uma interface de mais baixo nível (em C ou Fortran).

### PAPI Presets

É uma lista de eventos pré-definidos que é implementada nos sistemas que os suportam. Estes eventos oferecem contadores de desempenho.

Numa plataforma particular, cada evento pré-definido pode:

- (i) estar disponível diretamente como um contador simples,
- (ii) ser derivado através de uma combinação de contadores, ou
- (iii) não estar disponível.

### Saltos Condicionais:

Nome	Descrição
PAPI_BR_CN	Conditional branch instructions
PAPI_BR_INS	Branch instructions
PAPI_BR_MSP	Conditional branch instructions mispredicted
PAPI_BR_NTK	Conditional branch instructions not taken
PAPI_BR_PRC	Conditional branch instructions correctly predicted
PAPI_BR_TKN	Conditional branch instructions taken
PAPI_BR_UCN	Unconditional branch instructions
PAPI_BRU_IDL	Cycles branch units are idle
PAPI_BTAC_M	Branch target address cache misses

### Pedidos à Cache:

PAPI_CA_CLN	Requests for exclusive access to clean cache line
PAPI_CA_INV	Requests for cache line invalidation
PAPI_CA_ITV	Requests for cache line intervention
PAPI_CA_SHR	Requests for exclusive access to shared cache line
PAPI_CA_SNP	Requests for a snoop

### Store Condicional:

PAPI_CSR_FAL	Failed store conditional instructions
PAPI_CSR_SUC	Successful store conditional instructions
PAPI_CSR_TOT	Total store conditional instructions

### Operações de Vírgula Flutuante:

PAPI_FAD_INS	Floating point add instructions
PAPI_FDV_INS	Floating point divide instructions
PAPI_FMA_INS	FMA instructions completed
PAPI_FML_INS	Floating point multiply instructions
PAPI_FNV_INS	Floating point inverse instructions
PAPI_FP_INS	Floating point instructions
PAPI_FP_OPS	Floating point operations
PAPI_FP_STAL	Cycles the FP unit
PAPI_FPU_IDL	Cycles floating point units are idle
PAPI_FSQ_INS	Floating point square root instructions

### Contagem de Instruções:

PAPI_FUL_CCY	Cycles with maximum instructions completed
PAPI_FUL_ICY	Cycles with maximum instruction issue
PAPI_FXU_IDL	Cycles integer units are idle
PAPI_HW_INT	Hardware interrupts
PAPI_INT_INS	Integer instructions
PAPI_TOT_CYC	Total cycles
PAPI_TOT_IIS	Instructions issued
PAPI_TOT_INS	Instructions completed
PAPI_VEC_INS	Vector/SIMD instructions

### Acesso à Cache:

PAPI_L1_DCA	L1 data cache accesses
PAPI_L1_DCH	L1 data cache hits
PAPI_L1_DCM	L1 data cache misses
PAPI_L1_DCR	L1 data cache reads
PAPI_L1_DCW	L1 data cache writes
PAPI_L1_ICA	L1 instruction cache accesses
PAPI_L1_ICH	L1 instruction cache hits
PAPI_L1_ICM	L1 instruction cache misses
PAPI_L1_ICR	L1 instruction cache reads
PAPI_L1_ICW	L1 instruction cache writes
PAPI_L1_LDM	L1 load misses
PAPI_L1_STM	L1 store misses
PAPI_L1_TCA	L1 total cache accesses
PAPI_L1_TCH	L1 total cache hits
PAPI_L1_TCM	L1 total cache misses
PAPI_L1_TCR	L1 total cache reads
PAPI_L1_TCW	L1 total cache writes
PAPI_L2_DCA	L2 data cache accesses
PAPI_L2_DCH	L2 data cache hits
PAPI_L2_DCM	L2 data cache misses

PAPI_L2_DCR	L2 data cache reads
PAPI_L2_DCW	L2 data cache writes
PAPI_L2_ICA	L2 instruction cache accesses
PAPI_L2_ICH	L2 instruction cache hits
PAPI_L2_ICM	L2 instruction cache misses
PAPI_L2_ICR	L2 instruction cache reads
PAPI_L2_ICW	L2 instruction cache writes
PAPI_L2_LDM	L2 load misses
PAPI_L2_STM	L2 store misses
PAPI_L2_TCA	L2 total cache accesses
PAPI_L2_TCH	L2 total cache hits
PAPI_L2_TCM	L2 total cache misses
PAPI_L2_TCR	L2 total cache reads
PAPI_L2_TCW	L2 total cache writes
PAPI_L3_DCA	L3 data cache accesses
PAPI_L3_DCH	L3 Data Cache Hits
PAPI_L3_DCM	L3 data cache misses
PAPI_L3_DCR	L3 data cache reads
PAPI_L3_DCW	L3 data cache writes
PAPI_L3_ICA	L3 instruction cache accesses
PAPI_L3_ICH	L3 instruction cache hits
PAPI_L3_ICM	L3 instruction cache misses
PAPI_L3_ICR	L3 instruction cache reads
PAPI_L3_ICW	L3 instruction cache writes
PAPI_L3_LDM	L3 load misses
PAPI_L3_STM	L3 store misses
PAPI_L3_TCA	L3 total cache accesses
PAPI_L3_TCH	L3 total cache hits
PAPI_L3_TCM	L3 cache misses
PAPI_L3_TCR	L3 total cache reads
PAPI_L3_TCW	L3 total cache writes

### Acesso a Dados:

PAPI_LD_INS	Load instructions
PAPI_LST_INS	Load/store instructions completed
PAPI_LSU_IDL	Cycles load/store units are idle
PAPI_MEM_RCY	Cycles Stalled Waiting for memory Reads
PAPI_MEM_SCY	Cycles Stalled Waiting for memory accesses
PAPI_MEM_WCY	Cycles Stalled Waiting for memory writes
PAPI_PRF_DM	Data prefetch cache misses
PAPI_RES_STL	Cycles stalled on any resource
PAPI_SR_INS	Store instructions
PAPI_STL_CCY	Cycles with no instructions completed
PAPI_STL_ICY	Cycles with no instruction issue
PAPI_SYC_INS	Synchronization instructions completed

## Operações TLB (*Translation Lookaside Buffer*):

PAPI_TLB_DM	Data translation lookaside buffer misses
PAPI_TLB_IM	Instruction translation lookaside buffer misses
PAPI_TLB_SD	Translation lookaside buffer shutdowns
PAPI_TLB_TL	Total translation lookaside buffer misses

## Funções de Alto Nível do PAPI

**PAPI\_num\_counters** → devolve o número de contadores de *hardware* disponíveis.

**PAPI\_flips** → devolve o valor de Mflips/s (taxa de execução de instruções de vírgula flutuante).

**PAPI\_flops** → devolve o valor de Mflops / s (taxa de execução de operações de vírgula flutuante).

**PAPI\_ipc** → devolve o número de instruções por ciclo de relógio.

**PAPI\_accum\_counters** → adiciona os valores de contagem atuais ao conteúdo dum *array* e faz o *reset* dos contadores.

**PAPI\_read\_counters** → copia os valores de contagem atuais para um *array* e faz o *reset* dos contadores.

**PAPI\_start\_counters** → começa a contar os eventos de *hardware*.

**PAPI\_stop\_counters** → pára a contagem de eventos e devolve os valores atuais de contagem.

## Funções de Baixo Nível do PAPI

A extensa lista de funções de baixo nível do PAPI pode ser consultada na documentação incluída no ficheiro de instalação. Alguns exemplos:

- **PAPI\_perror** converte os códigos de erro do PAPI para *strings* e envia a mensagem de erro para *stderr*.
- **PAPI\_library\_init** inicializa utilização da biblioteca PAPI. Deve ser chamada antes de se usar qualquer função de baixo nível. Devolve o estado da biblioteca PAPI: em caso de sucesso devolve o valor PAPI\_VER\_CURRENT.
- **PAPI\_shutdown** termina a utilização do PAPI e liberta todos os recursos e a memória utilizada pelo PAPI.
- **PAPI\_get\_real\_usec** devolve o tempo real, em  $\mu$ s, desde um ponto inicial arbitrário (normalmente desde o arranque da máquina).

## Comandos do PAPI

**papi\_avail** → indica a disponibilidade e informação detalhadas sobre os eventos pré-definidos do PAPI.

**papi\_clockres** → mede e fornece informação sobre a latência e a resolução dos temporizadores do PAPI.

**papi\_cost** → calcula o tempo de execução das operações básicas do PAPI (pares *start/stop* e *read's*).

**papi\_command\_line** → executa eventos do PAPI, pré-definidos ou nativos, a partir da linha de comando.

**papi\_decode** → converte a lista de eventos pré-definidos do PAPI para um formato CSV, que pode ser facilmente lido e alterado por qualquer folha de cálculo ou editor de texto; o resultado pode também ser fornecido à função *PAPI\_encode\_events()*.

**papi\_event\_chooser** → dada uma lista de eventos, devolve os eventos que podem ser contados com base nessa lista.

**papi\_mem\_info** → fornece informação sobre a arquitetura de memória do processador que estamos a usar.

**papi\_native\_avail** → fornece informação detalhada sobre os eventos nativos do PAPI.