Библиотека PAPI

2021

PAPI

PAPI — Performance Application Programming Interface.

http://icl.cs.utk.edu/papi/

Установка

Debian GNU/Linux 11:

```
# apt-get install libpapi-dev libpapi6.0 papi-tools
```

```
$ git clone https://bitbucket.org/icl/papi.git
$ cd papi/src
$ ./configure [--prefix=/path/to/papi]
$ make
$ make install
```

PAPI. Использование.

```
# echo 0 > /proc/sys/kernel/perf_event_paranoid
```

Получение информации о доступных аппартных счётчиках:

```
$ papi_avail

Available PAPI preset and user defined events plus hardware information.
--------------------------------------------------------------------------------
PAPI version             : 6.0.0.0
Operating system         : Linux 5.10.0-8-amd64

...

================================================================================
  PAPI Preset Events
================================================================================
    Name        Code    Avail Deriv Description (Note)
PAPI_L1_DCM  0x80000000  Yes   No   Level 1 data cache misses
PAPI_L1_ICM  0x80000001  Yes   No   Level 1 instruction cache misses
PAPI_L2_DCM  0x80000002  Yes   Yes  Level 2 data cache misses
PAPI_L2_ICM  0x80000003  Yes   No   Level 2 instruction cache misses
PAPI_L3_DCM  0x80000004  No    No   Level 3 data cache misses
PAPI_L3_ICM  0x80000005  No    No   Level 3 instruction cache misses
PAPI_L1_TCM  0x80000006  Yes   Yes  Level 1 cache misses
PAPI_L2_TCM  0x80000007  Yes   No   Level 2 cache misses
PAPI_L3_TCM  0x80000008  Yes   No   Level 3 cache misses

...
```

PAPI. Использование.

В исходном коде:

```
#include <papi.h>
```

Компиляция:

```
gcc example.c -lpapi -o example
-I/path/to/papi/include -L/path/to/papi/lib
```

PAPI. Инициализация.

```
int PAPI_library_init(int version);
```

```
int
main(int argc, char **argv)
{
     ...
     if (PAPI_library_init(PAPI_VER_CURRENT) != PAPI_VER_CURRENT)
        exit(1);
     ...
}
```

PAPI. Получение информации об ошибке.

```
#include <stdlib.h>
#include <stdio.h>
#include <papi.h>

void handle_error (int retval)
{
     printf("PAPI error %d: %s\n", retval, PAPI_strerror(retval));
     exit(1);
}
```

PAPI. Наборы событий (**event sets**)

```
int PAPI_create_eventset( int * EventSet );
int PAPI_add_event( int  EventSet, int  EventCode );
```

Примеры кодов событий: `PAPI_TOT_INS`, `PAPI_L1_TCM`.

```
main()
{
    int retval, EventSet=PAPI_NULL;

    /* Initialize the PAPI library */
    retval = PAPI_library_init(PAPI_VER_CURRENT);
    if (retval != PAPI_VER_CURRENT) {
        fprintf(stderr, "PAPI library init error!\n");
        exit(1);
    }

    /* Create the Event Set */
    if (PAPI_create_eventset(&EventSet) != PAPI_OK)
        handle_error(1);

    /* Add Total Instructions Executed to our Event Set */
    if (PAPI_add_event(EventSet, PAPI_TOT_INS) != PAPI_OK)
        handle_error(1);

    ...
```

PAPI. Измерение количества событий

```
int PAPI_start( int EventSet );
int PAPI_read(int EventSet, long_long * values );
int PAPI_reset( int EventSet );
int PAPI_accum( int  EventSet, long_long * values );
int PAPI_stop( int EventSet, long long * values );
```

```
long_long values[1];

...

/* Start counting events in the Event Set */
if (PAPI_start(EventSet) != PAPI_OK)
    handle_error(1);

...
/* SOME COMPUTATIONAL CODE */
...

/* Read the counting events in the Event Set */
if (PAPI_read(EventSet, values) != PAPI_OK)
    handle_error(1);
```

PAPI. Упрощённые функции.

```
int PAPI_flops_rate ( int event, float *rtime, float *ptime,
                      long long *flpops, float *mflops );

int PAPI_flips_rate( int event, float *rtime, float *ptime,
                     long long *flpins, float *mflips );

int PAPI_epc( int event, float *rtime, float *ptime, long long *ref,
              long long *core, long long *evt, float *epc );

int PAPI_ipc( float *rtime, float *ptime, long long *ins, float *ipc );

int PAPI_rate_stop();
```

PAPI. Упрощённые функции.

```
if ( (retval = PAPI_flops_rate(PAPI_FP_OPS, &real_time, &proc_time, &flpops, &mflops)) < PAPI_OK )
  {
    printf("Could not initialise PAPI_flops \n");
    printf("Your platform may not support floating point operation event.\n");
    printf("retval: %d\n", retval);
    exit(1);
  }

  your_slow_code();


  if ( (retval = PAPI_flops_rate(PAPI_FP_OPS, &real_time, &proc_time, &flpops, &mflops)) < PAPI_OK )
  {
    printf("retval: %d\n", retval);
    exit(1);
  }


  printf("Real_time: %f Proc_time: %f flpops: %lld MFLOPS: %f\n",
         real_time, proc_time, flpops, mflops);
```