

<http://itpro.nikkeibp.co.jp/article/COLUMN/20061102/252525/>
<http://apis.jpn.ph/fswiki/wiki.cgi?page=Java%2Fhprof%28jvmpi%29>

hprof

■ とりあえず使ってみる

```
java -agentlib:hprof -jar Java2Demo.jar
```

デフォルトではプロファイル結果は java.hprof.txt になる。

■ 結果を見やすく

```
java -agentlib:hprof=format=bJava2Demo.jar
```

format=b で結果をバイナリ化する。結果ファイルは java.hprof になる。

```
jhat java.hprof
```

で結果を解析して web ブラウザで見れるようになる。

<http://localhost:7000>

で見れる。

ただし、今のところヒープしか対応していないみたい。

cpu に関する情報は、テキストで見ると見るしかないようだ。

結果ファイルの見方

■ ヒーププロファイリング

SITES BEGIN から SITES END の部分が、使用メモリの統計情報で、メモリの使用量順に出力される。各カラム内容は以下の通り。

タイトル	意味
rank	消費メモリの順位
percent self	消費メモリのパーセンテージ
percent accum	消費メモリの累積パーセンテージ (上位 rank の percent self との和)
live bytes	ライブオブジェクト (GC されていないオブジェクト) の消費バイト数
live objs	ライブオブジェクトのオブジェクト数
alloc'ed bytes	アロケートされたオブジェクト (GC されたオブジェクトを含む) の消費バイト数

alloc'ed objs	アロケートされたオブジェクトのオブジェクト数
stack trace	対応するトレース番号
class name	クラス名

クラス名の部分に記述されている [C や [I といったクラス名は配列を表す。

クラス名	意味
[L クラス名	オブジェクトの配列
[Z	boolean 型の配列
[B	byte 型の配列
[C	char 型の配列
[S	short 型の配列
[I	int 型の配列
[J	long 型の配列
[F	float 型の配列
[D	double 型の配列

■ CPU 使用率プロファイリング

CPU TIME (ms) BEGIN から CPU TIME (ms) END の部分が、CPU 使用率の情報で、使用率順に出力される。各カラム内容は以下の通り。

タイトル	意味
rank	CPU 時間の順位
self	CPU 時間のパーセンテージ
accum	CPU 使用時間の累積パーセンテージ (上位 rank の self との和)
count	ヒットしたサンプリング数 (samples 指定) / メソッドのコール回数 (times 指定)
trace	対応するトレース番号
method	メソッド名

オプション

オプション名前と値	オプションの説明	デフォルト
heap=dump sites all	ヒープのプロファイリング	all
cpu=samples times old	CPU 使用率のプロファイル	off
monitor=y n	モニターの競合	n
format=a b	ASCII またはバイナリ形式	a
file=name	データをファイルに書き出す	java.hprof[.txt]

net=host:port	データをソケット経由で送信	ファイルに書き出す
depth=size	スタックトレースの深さ	4
cutoff=value	出力のカットオフポイント	0.0001
lineno=y n	トレースに行番号を出力するか	y
thread=y n	トレースにスレッドを出力するか	n
doe=y n	終了時にダンプするか	y
gc_okay=y n	GC okay during sampling	y

■ ヒーププロファイリングのオプション値

引数	動作
sites	使用メモリの統計情報を出力
dump	ヒープのダンプを出力
all	sites と dump の両方を出力

■ CPU 使用率プロファイルのオプション値

引数	動作
samples	定期的にサンプリングし、アクティブなスレッド（メソッド）をカウントする
times	メソッドの IN/OUT を監視し、メソッドのコール回数をカウントする
old	times と同様。出力形式が、旧 hprof フォーマットとなる