

C リソースと制限

[Programming C]

Linux system で実行される program には、resource に関する種々の制限が課せられている。

- Hardware による物理的な制限 (memory など)
- System policy による制限 (CPU 時間の上限など)
- 実装上の制限 (int の size、file 名に使用できる文字数など)

このうちいくつかは UNIX の仕様で定義されており、application から取得することができる。
header file limits.h では operating system によって課せられた制約を表すための定数が定義されている。

定数	説明
NAME_MAX	file 名に使用できる最大文字数
CHAR_BIT	char の bit 数
CHAR_MAX	char の最大値
INT_MAX	int の最大値

header file sys/resource.h には、resource の操作に関する様々な定義が記述されている。

```
#include <sys/resource.h>
int getpriority(int which, id_t who);
int setpriority(int which, id_t who, int priority);
int getrlimit(int resource, struct rlimit *r_limit);
int setrlimit(int resource, const struct rlimit *r_limit);
int getrusage(int who, struct rusage *r_usage);
```

- id_t は、user id と group id に使われる整数型
- 構造体 rusage は現在の program がどれだけの CPU 時間を使用しているかを取得

rusage 構造体の member

member	説明
struct timeval ru_utime	使用した user 時間
struct timeval ru_stime	使用した system 時間

parameter who には、次の何れかの定数

定数	説明
RUSAGE_SELF	現在の program に関する使用情報のみを返す
RUSAGE_CHILDREN	子 process に関する使用情報も含める

- application では getpriority、setpriority 関数を使用して、program の優先度を調べたり、設定したりできる。
- これらの関数を使って優先度を取得、変更する process は、process id、group id、user の何れかで指定。
- parameter who をどのように扱うのか指定するのが、parameter which

parameter which には次の何れかの定数

定数	説明
PRIO_PROCESS	who は process id
PRIO_PGRP	who は process group
PRIO_PUSER	who は user

- ・ setpriority 関数は、可能であれば、新しい優先度を設定
 - ・ default の優先度は 0
 - ・ 正の優先度は、background task
 - ・ 負の優先度は、より頻繁に実行され、CPU 時間も多く使用
 - ・ 有効な範囲は -20 ~ +20 数値が大きいほど優先度は低い
-
- ・ system resource に関する制限を取得、設定するには、getrlimit、setrlimit を使う
 - ・ resource の制限を表すために、構造体 rlimit を使用する。

rlimit 構造体 (sys/resource.h で定義)

member	説明
rlim_t rlim_cur	現在の soft 制限
rlim_t rlim_max	hard 制限

通常 soft 制限は、超えるべきではない制限を意味する。

hard 制限を越えた場合は、system が program に signal を送り program を終了させようとする

rlimit 関数群の parameter resource で指定する

parameter resource

parameter	説明
RLIMIT_CORE	core dump file size の制限 (bite 単位)
RLIMIT_CPU	CPU 時間の制限 (秒単位)
RLIMIT_DATA	data(malloc/sbrk)segment の制限 (bite 単位)
RLIMIT_FSIZE	file size の制限 (bite 単位)
RLIMIT_NOFILE	open する file 数の制限
RLIMIT_STACK	stack size の制限 (bite 単位)
RLIMIT_AS	address 空間 (stack および data) の制限 (bite 単位)

```
#include <sys/resource.h>
#include <stdio.h>
#include <stdlib.h>
int main()
{
    struct rusage r_usage;
    int priority = getpriority(PRIO_PROCESS, getpid());
    printf("Priority : %d\n", priority);

    /* dummy work */
}
```

```
int i;
for(i=0; i<30000; i++) {
    getpriority(PRIO_PROCESS, getpid());
}
getrusage(RUSAGE_SELF, &r_usage);
printf("CPU usage: User=%Td.%06ld, System=%Td.%06ld\n"
    , r_usage.ru_utime.tv_sec, r_usage.ru_utime.tv_usec
    , r_usage.ru_stime.tv_sec, r_usage.ru_stime.tv_usec);
exit(0);
}
```

この本からの覚書。