

共有メモリ(Shared memory)について

神田

共有メモリ：複数のプロセスでメモリを共有する仕組み。

ハードウェア的なやりかたと、ソフトウェア的なやりかたがある。

例えば、DGSでは、ハードウェア的に、複数のマシンの共有メモリ（ボード）を連結している。

DMGのデータ転送系では、ソフトウェア的に、同一ホスト上の複数のプロセスで共有する。

参考webサイト

<http://ja.wikipedia.org/wiki/共有メモリ>

<http://blog.majide.com/2009/03/c-shared-memory-program/>

<http://www.ncad.co.jp/~komata/c-kouza13.htm>

など

今日は、<http://blog.majide.com/2009/03/c-shared-memory-program/> にある
サンプルプログラムを使ってみます。

システムコール	説明
<code>shmget()</code>	共有メモリ・セグメント識別子を獲得する
<code>shmat()</code>	自プロセスのデータセグメントにマップ(アタッチとも呼ぶ)する
<code>shmdt()</code>	共有メモリをアンマップ(デタッチとも呼ぶ)する
<code>shmctl()</code>	共有メモリをシステム上から削除する

共有メモリを読むプログラム shm_reciver.c

```
15a1' 528æM-^W¥ 12:23      shm_reciver.c      1/1 ãM-^CM-^ZãM-^C¼ãM-^B,
#include <stdio.h>
#include <stdlib.h>
#include <string.h> /* strcpy */
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
#include <unistd.h> /* explicit definition of sleep */

int main(void)
{
    int id;
    char *adr;

    /* IPC_PRIVATEを指定して共有メモリを作成する */
    if((id = shmget(IPC_PRIVATE, 512, IPC_CREAT|0666)) == -1){
        perror("shmget");
        exit(-1);
    }

    printf("共有メモリID = %d\n", id);
}
```

```

printf("ID = %d\n", id);

/* char *shmat */
if(( adr = (char *)shmat(id, NULL, 0)) == (void *)-1){
    perror("shmat");
} else {
    strcpy(adr, "Initial");
    /* Initial */
    while(1){
        printf("%s\n", adr);
        /* iÖendi×Ç½âî» */
        if (strcmp(adr, "end") == 0) {
            break;
        }
        sleep(1);
    }

    if (shmdt(adr) == -1) {
        perror("shmdt");
    }
}

/* shmÇÊ²öð×1ë */
if(shmctl(id, IPC_RMID, 0) == -1){
    perror("shmctl");
    exit(EXIT_FAILURE);
}

return 0;
}
}

```

共有メモリに書き込むプログラム shm_writer.c

```
15a1' 528æM-^W¥ 12:23      shm_writer.c      1/1 äM-^CM-^ZäM-^C¼äM-^B,  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h> /* strcpy */  
#include <sys/types.h>  
#include <sys/ipc.h>  
#include <sys/shm.h>  
  
int main(int argc, char *argv[])  
{  
    int    id;  
    char  *adr;  
  
    if( argc <= 2) {  
        fprintf(stderr, "Usage: shm_writer shm_id string\n");  
        exit(EXIT_FAILURE);  
    }  
  
    id = atoi(argv[1]);  
  
    if(( adr = (char *)shmat(id,0,0)) == (void *)-1) {  
        perror("shmat");  
    } else {  
        strcpy(adr, argv[2]);  
        fprintf(stderr, "written.\n");  
        if( shmdt(adr) == -1) {  
            perror("shmdt");  
        }  
    }  
}
```

Pleiades-01 上でテスト

```
ttys001 - 80x24
-bash-4.1$ emacs shm_reciver.c

[1]+ 停止                emacs shm_reciver.c
-bash-4.1$ gcc shm_reciver.c -o shm_reciver
-bash-4.1$ ./shm_reciver
共有メモリ ID = 753665
Initial
Initial
Initial
Initial
hoge
hoge
hoge
KAGRA
KAGRA
KAGRA
KAGRA
2015年
2015年
2015年
2015年
2015年
end
-bash-4.1$ fg
```

```
ttys000 - 80x24
shared_mem.zip          100% 4814    4.7KB/s    00:00
50 ~/src > ssh taurus-01
kanda@taurus-01's password:
Last login: Thu May 28 11:40:02 2015 from 192.168.30.205
-bash-4.1$ ssh pleiades-01
kanda@pleiades-01's password:
Last login: Thu May 28 11:40:19 2015 from taurus-01
-bash-4.1$ cd src/
-bash-4.1$ ls
shared_mem  shared_mem.zip
-bash-4.1$ cd shared_mem
-bash-4.1$ ls
shm_reciver  shm_reciver.c  shm_reciver.c~  shm_writer  shm_writer.c
-bash-4.1$ ./shm_writer 753665 hoge
written.
-bash-4.1$ ./shm_writer 753665 KAGRA
written.
-bash-4.1$ ./shm_writer 753665 `date`
written.
-bash-4.1$ date
2015年  5月 28日 木曜日 11:44:50 JST
-bash-4.1$ ./shm_writer 753665 end
written.
-bash-4.1$
```