

Subject: Lecture on Operating System Lecture Exercise 5.2

From: IKENOYA Katsutoshi <j05002@ie.u-ryukyu.ac.jp>

Date: Wed, 07 Feb 2007 16:09:53 +0900

To: Shinji KONO <kono@ie.u-ryukyu.ac.jp>

学籍番号 : 055702B

- ・ 2相ロックでない場合を修正しました。

問題5.2

デッドロックを2相ロックで再現してみよ。
また、2相ロックでない場合に矛盾が生じる例を示せ。

今回作成したソースを以下の場所に置く。

```
/home/y05/j05002/OS/flock_test/flock_test1.pl  
/home/y05/j05002/OS/flock_test/flock_test2.pl  
/home/y05/j05002/OS/flock_test/flock_test3.pl  
/home/y05/j05002/OS/flock_test/flock_test4.pl  
-----
```

(1)2相ロック

以下の条件でデッドロックを再現する。

- 1 (flock_test1.pl) : PLANE -> HOTEL の順に予約
- 2 (flock_test1.pl) : HOTEL -> PLANE の順に予約

- ・ ソース

- flock_test1.pl

省略

```
flock(PLANE,LOCK_EX);  
sleep 1;  
&reserve('PLANE',$plane);
```

```
flock(HOTEL,LOCK_EX);  
sleep 1;  
&reserve('HOTEL',$hotel);
```

```
close(PLANE); # and release lock  
close(HOTEL); # and release lock  
省略
```

- flock_test2.pl

省略

```
flock(HOTEL,LOCK_SH);  
sleep 1;  
&reserve('HOTEL',$hotel);
```

```
flock(PLANE,LOCK_SH);  
sleep 1;  
&reserve('PLANE',$plane);
```

```
close(HOTEL); # and release lock  
close(PLANE); # and release lock  
省略
```

・実行結果

```
[j05002@flock_test]% perl flock_test1.pl & perl flock_test2.pl
[1] 8258
1:plane_reserve reserved
2:hotel_reserve reserved
^C <-デッドロックが発生
[j05002@flock_test]% 1:hotel_reserve reserved

[1] Done perl flock_test1.pl
```

1のプロセスがPLANE, 2のプロセスがHOTELを予約してアンロックの処理を実行しようとする、1の場合は2がHOTELをロックしており、2の場合は1がPLANEをロックしているので、デッドロックが発生する。

(2)2相ロックでない場合に矛盾が生じる例

予約の順番を以下のようにする。

```
1 (flock_test3.pl) : PLANE -> HOTEL の順に予約
2 (flock_test4.pl) : HOTEL -> PLANE の順に予約
```

・ソース

- flock_test3.pl

省略

```
flock(PLANE,LOCK_SH);
sleep 1;
&reserve('PLANE',$plane);

close(PLANE); # and release lock

flock(HOTEL,LOCK_SH);
sleep 1;
&reserve('HOTEL',$hotel);

close(HOTEL); # and release lock
```

省略

- flock_test4.pl

省略

```
flock(HOTEL,LOCK_SH);
sleep 1;
&reserve('HOTEL',$hotel);

close(HOTEL); # and release lock

flock(PLANE,LOCK_SH);
sleep 1;
&reserve('PLANE',$plane);

close(PLANE); # and release lock
```

省略

・実行結果

```
[j05002@flock_test]% cat hotel_reserve plane_reserve
1
1
[j05002@flock_test]% perl flock_test3.pl & perl flock_test4.pl
[1] 25530
3:hotel_reserve reserved
4:hotel_reserve reserved
3:25530 can't reserve plane_reserve
4:25531 can't reserve plane_reserve
[1] + Exit 255 perl flock_test3.pl
[j05002@flock_test]% cat hotel_reserve plane_reserve
0
0
```

本来ならば、HOTELの空きもPLANEの空きも1なので、4のプロセスはPLANEを予約できて、HOTELは予約できないという結果になるべきだが、実行結果を見るとHOTELを予約できて、PLANEは予約できていない。さらに実行後のHOTELの空きとPLANEの空きが共に0になっている。以上の事から矛盾が起きている事が分かる。