

OS-Project6

常烁晨-521021910369

Task1:初始化

初始化全局变量，available数组从命令行读取，allocation矩阵初始为0，maximum和need矩阵从input.txt文件读取。

代码介绍：

```
int main(int argc, char* argv[])
{
    for(int i = 0; i < NUMBER_OF_RESOURCES; i++)
        available[i] = atoi(argv[i+1]);
    FILE *input;
    input = fopen("input.txt", "r");
    char *line = (char*)malloc(sizeof(char)*100);
    for(int i = 0; i < NUMBER_OF_CUSTOMERS; i++)
    {
        // deal with the input file. 略
    }
}
```

通过命令行读入数组 available，通过文件读写，读入数组max

Task2:打印指令

指令*打印 available, maximum, allocation 和 need 数组的值。

代码介绍：

```
if(strcmp(command, "*\n") == 0)
{
    printf("Available:\n"); //略
    printf("Maximum:\n"); //略
    printf("Allocation:\n"); //略
    printf("Need:\n"); //略
    continue;
}
```

Task3: 请求资源

请求资源时，首先检查请求的资源是否超出目前可用/超出进程所需最大要求，若正常，则执行银行家算法，检查是否处于安全状态。

银行家算法的步骤简述为：

(1) 首先分配资源给该进程。(2) 初始化 finish 数组为 0，temp 数组为分配资源后的 available。(3) 依次判断进程 1 ~ n 中 finish 为 0 的进程所对应的数组 need，若某进程 $need < temp$ ，则将其 finish 设置为 1，并将 $temp += allocation$ ，直到没有进程满足。(4) 检查此时是否所有进程都已完成。

若所有进程都已完成，则判断该请求安全，否则被证明是不安全的。由于篇幅原因，代码较长，因此不再在报告中展示。

Task4: 释放资源

由于释放资源不会导致当前状态由安全转变为不安全，因此只需检查释放资源是否超出当前进程所占有的资源即可。

```
void release_resources(int customer_num, int
release[])
{
//check whether exceeding!
    for(int i = 0; i < NUMBER_OF_RESOURCES; i++)
    {
        available[i] += release[i];
        allocation[customer_num][i] -= release[i];
        need[customer_num][i] += release[i];
    }
    printf("Successfully release the resources!\n");
}
```

以下简单展示实验运行结果。

```
parallels@ubuntu-linux-22-04-desktop:~/final-src-osc10e/ch8$ ./banker 10 6 7 8
Enter Command:
*
Available:
10 6 7 8
Maximum:
6 4 7 3
4 2 3 2
2 5 3 3
6 3 3 2
5 6 7 5
Allocation:
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
Need:
6 4 7 3
4 2 3 2
2 5 3 3
6 3 3 2
5 6 7 5

Enter Command:
RQ 0 1 1 1 1
Successfully allocate the resources!
Request Granted.

Enter Command:
RQ 1 4 2 3 3
1 customer doesn't need so many resources!
Request Denied.

Enter Command:
RQ 1 4 2 3 2
Successfully allocate the resources!
Request Granted.

Enter Command:
RL 0 2 1 1 1
0 customer doesn't have so many resources!
Resources Released.

Enter Command:
*
Available:
5 3 3 5
Maximum:
6 4 7 3
4 2 3 2
2 5 3 3
6 3 3 2
5 6 7 5
Allocation:
1 1 1 1
4 2 3 2
0 0 0 0
0 0 0 0
0 0 0 0
Need:
5 3 6 2
0 0 0 0
2 5 3 3
6 3 3 2
5 6 7 5

Enter Command:
exit
parallels@ubuntu-linux-22-04-desktop:~/final-src-osc10e/ch8$
```