os 作业 2

银行家问题

```
#include <stdio.h>
  int curr[5][5], maxclaim[5][5], avl[5];
  int alloc[5] = {0, 0, 0, 0, 0};
  int maxres[5], running[5], safe=0;
  int count = 0, i, j, exec, r, p, k = 1;

  int main()
    {
        printf("\nEnter the number of processes: ");
        scanf("%d", &p);

        for (i = 0; i < p; i++) {
            running[i] = 1;
            count++;
        }

        printf("\nEnter the number of resources: ");
        scanf("%d", &r);

        printf("\nEnter Claim Vector:");
        for (i = 0; i < r; i++) {
            scanf("%d", &maxres[i]);
        }

        printf("\nEnter Allocated Resource Table:\n");
        for (i = 0; i < p; i++) {
            for(j = 0; j < r; j++) {
                scanf("%d", &curr[i][j]);
            }
        }

        printf("\nEnter Maximum Claim Table:\n");
        for (i = 0; i < p; i++) {
            for(j = 0; j < r; j++) {
                scanf("%d", &maxclaim[i][j]);
            }
        }
```

```
printf("\nThe Claim Vector is: ");
for (i = 0; i < r; i++) {
        printf("\t%d", maxres[i]);
}

printf("\nThe Allocated Resource Table:\n");
for (i = 0; i < p; i++) {
        for (j = 0; j < r; j++) {
                printf("\t%d", curr[i][j]);
        }

        printf("\n");
}

printf("\nThe Maximum Claim Table:\n");
for (i = 0; i < p; i++) {
        for (j = 0; j < r; j++) {
                printf("\t%d", maxclaim[i][j]);
        }

        printf("\n");
}

for (i = 0; i < p; i++) {
        for (j = 0; j < r; j++) {
                alloc[j] += curr[i][j];
        }
}

printf("\nAllocated resources:");
for (i = 0; i < r; i++) {
        printf("\t%d", alloc[i]);
}

for (i = 0; i < r; i++) {
        avl[i] = maxres[i] - alloc[i];
}

printf("\nAvailable resources:");
for (i = 0; i < r; i++) {
        printf("\t%d", avl[i]);
}
printf("\n");
```

```
    //Main procedure goes below to check for unsafe state.
    while (count != 0) {
        safe = 0;
        for (i = 0; i < p; i++) {
            if (running[i]) {
                exec = 1;
                for (j = 0; j < r; j++) {
                    if (maxclaim[i][j] - curr[i][j] > avl[j]) {
                        exec = 0;
                        break;
                    }
                }
                if (exec) {
                    printf("\nProcess%d is executing\n", i + 1);
                    running[i] = 0;
                    count--;
                    safe = 1;

                    for (j = 0; j < r; j++) {
                        avl[j] += curr[i][j];
                    }

                    break;
                }
            }
        }
        if (!safe) {
            printf("\nThe processes are in unsafe state.\n");
            break;
        } else {
            printf("\nThe process is in safe state");
            printf("\nAvailable vector:");

            for (i = 0; i < r; i++) {
                printf("\t%d", avl[i]);
            }

            printf("\n");
        }
    }
}
```