

オペレーティングシステム

第9回ミニレポート課題

提出者名:小林 亮太

学籍番号:09B19025

提出年月日:令和3年5月14日

1 解答

空間的参照局所性のある、行列の積を計算するCプログラムを以下に示す。

```
#include<stdio.h>

int main()
{
    int i, j,k;
    int A[4][4] = { {1,2,0,0},
                    {0,0,2,1},
                    {1,2,2,0},
                    {0,1,0,0} };

    int B[4][4] = { {2,0,0,0},
                    {0,1,1,1},
                    {0,1,1,0},
                    {1,0,2,2} };

    int C[4][4] = { {0,0,0,0},
                    {0,0,0,0},
                    {0,0,0,0},
                    {0,0,0,0} };

    for ( i = 0; i < 4; i++)
    {
        for ( k = 0; k < 4; k++)
        {
```

```

for ( j = 0; j < 4; j++)
{
C[i][j] = C[i][j] + A[i][k] * B[k][j];
}
}
}

```

```

for ( i = 0; i < 4; i++)
{
for ( j = 0; j < 4; j++)
{
printf("%d ", C[i][j]);
}
printf("\n");
}
}

```

C 言語では、同じ行の要素がメモリ上に連続して配置される。そのため、同じ行の要素に続けてアクセスした方が使用するメモリアドレスも連続的になり、空間的参照局所性が高くなる。j は行列 B,C の列のインデックスであるため、最も内側のループにすることで、空間的参照局所性を高めている。

時間的参照局所性のある、行列の積を計算する C プログラムを以下に示す。

```

#include<stdio.h>

int main()
{
int i, j, k;
int p, q, r;
int A[4][4] = { {1,2,0,0},
{0,0,2,1},
{1,2,2,0},
{0,1,0,0} };

int B[4][4] = { {2,0,0,0},
{0,1,1,1},
{0,1,1,0},
{1,0,2,2} };

int C[4][4] = { {0,0,0,0},
{0,0,0,0},
{0,0,0,0},

```

```
{0,0,0,0} };
```

```
for ( p = 0; p < 4; p = p + 2)
{
for (r = 0; r < 4; r = r + 2)
{
for ( q = 0; q < 4; q = q + 2)
{
for (i = p; i < p+2; i++)
{
for (k = r; k < r+2; k++)
{
for (j = q; j < q+2; j++)
{
C[i][j] = C[i][j] + A[i][k] * B[k][j];
}
}
}
}
}
}
```

```
for (i = 0; i < 4; i++)
{
for (j = 0; j < 4; j++)
{
printf("%d ", C[i][j]);
}
printf("\n");
}
}
```

4×4の行列を2×2の4つの行列に分割して計算することにより、短い時間に同じ部分行列を複数回参照できるになっている。これにより、時間的参照局所性を高めている。