

Katedra Inżynierii Komputerowej
Politechnika Częstochowska

Laboratorium
Programowania niskopoziomowego

Laboratorium 6

dr inż. Dziwiński Piotr

22 listopada 2011

1 Suma macierzy dwuwymiarowych

Proszę napisać funkcję w assemblerze sumującą dwie macierze dwuwymiarowe zgodnie ze wzorem:

$$w_{ij} = u_{ij} + v_{ij} \quad (1)$$

Przykładowy kod C++ realizujący to zadanie przedstawia się następująco:

```
int ** suma_mac(int **u, int **v, int **w, int n1, int n2)
{
    for(int i=0; i<n1; i++)
    {
        for(int j=0; j<n2; j++)
        {
            w[i][j] = u[i][j]+v[i][j];
        }
    }
    return w;
}
```

2 Poszukiwanie elementu minimalnego w macierzy trójwymiarowej

Proszę napisać funkcję w assemblerze znajdującą wartość minimalną. Przykładowy kod C++ realizujący to zadanie może wyglądać następująco:

```
int min_matrix3D(int *** tab, int n1, int n2, int n3)
{
    int min=tab[n1-1][n2-1][n3-1];
    for(int i=0; i<n1; i++)
        for(int j=0; j<n2; j++)
            for(int k=0; k<n3; k++)
                if (min>tab[i][j][k]) min = tab[i][j][k];
    return min;
}
```

Wielokrotnie zagnieżdżone pętle można zaimplementować w assemblerze następująco:

```
int min_matrix3D_asm(int *** tab, int n1, int n2, int n3)
{
    __asm
    {
        mov ecx, n1;
L1:
        push ecx;                //stos {i} i = [esp]
        mov ecx, n2;
L2:
        push ecx;                //stos {i,j} i = [esp-4], j = [esp]
        mov ecx, n3;
L3:

        Loop L3;
```

```

        pop ecx;                //stos{i}

        Loop L2;
        pop ecx;                //stos{empty}
        Loop L1;
    }
}

```

Powyższy kod należy uzupełnić tak, aby realizował to samo zadanie co wcześniejszy kod C++.

3 Mnożenie dwóch macierzy

Proszę napisać metodę mnożącą macierz $\mathbf{M}_{N_1 \times N_2}$ i macierz $\mathbf{V}_{N_2 \times N_3}$ zgodnie z następującym wzorem:

$$u_{n_1 n_3} = \sum_{n_2=0}^{N_2-1} m_{n_1 n_2} \cdot v_{n_2 n_3} \quad (2)$$

gdzie: N_1 - liczba wierszy macierzy \mathbf{M} , N_2 - liczba kolumn macierzy \mathbf{M} , N_3 - liczba kolumn macierzy \mathbf{V} .

Operację mnożenia należy zaimplementować w C++ oraz w asemblerze.

Rozwiązanie (implementacja C++, wskaźniki):

```

void iloczyn(int **m, int **v, int **u, int N1, int N2, int N3)
{
    for(int n1=0; n1<N1; n1++)
        for(int n3=0; n3<N3; n3++)
        {
            u[n1][n3] = 0;
            for(int n2=0; n2<N2; n2++)
                u[n1][n3] = u[n1][n3] + m[n1][n2] * v[n2][n3];
        }
}

```

Rozwiązanie (implementacja asembler, wskaźniki):

```

void iloczyn_asm(int **m, int **v, int **u, int N1, int N2, int N3)
{
    __asm {
        mov ecx, N1;           //pętla zewnętrzna n1
for11:

        push ecx;             //stos{n1} n1 = [esp]
        mov ecx, N3;          //licznik dla n3
for33:

        mov ebx, 0;           //u[n1][n3]=0; (ebx)
        mov esi, m;
        mov eax, [esp];       //n1 -> eax
        mov esi, [esi+4*eax-4]; //m[n1]

        push ecx;             //stos{n1, n3}, n1 = [esp-4], n3 = [esp]
        mov ecx, N2;          //licznik dla n2
for22:

        mov edi, v;
        mov eax, [esp];       //stos{n1, n3}, n3 -> eax
        mov edi, [edi+4*ecx-4]; //v[n2]
    }
}

```

```

        mov eax, [edi + 4*eax - 4]; //v[n2][n3]
        mul [esi + 4*ecx - 4]; //m[n1][n2]*v[n2][n3]
        add ebx, eax; //ebx+=m[n1][n2]*v[n2][n3]
loop for22;
        pop ecx; //ecx <- n3, stos{n1}
        mov esi, u;
        mov eax, [esp]; //stos{n1}, n1 -> eax
        mov esi, [esi + 4*eax - 4]; //u[n1]
        mov [esi + 4*ecx - 4], ebx; //u[n1][n3]=ebx;
loop for33;
        pop ecx; //n1 -> ecx, stos{empty}
loop for11;
}}

```

Rozwiązanie (implementacja C++, tablice statyczne):

```

void iloczyn(int m[][5], int v[][5], int u[][5], int N)
{
    for(int n1=0; n1<N; n1++)
        for(int n3=0; n3<N; n3++)
        {
            u[n1][n3] = 0;
            for(int n2=0; n2<N; n2++)
                u[n1][n3] = u[n1][n3] + m[n1][n2] * v[n2][n3];
        }
}

```

Proszę zaimplementować mnożenie macierzy dla tablic statycznych.