

quieres trabajar
en Wuolah??

TE BUSCAMOS

28/5/2021

SWAD: plataforma de apoyo a la docencia / UGR /

?

Examen BP4 - Grupo B?



Universidad de Granada - Grado en Ingeniería Informática
Arquitectura de Computadores



Desconocido:

sin ánimo
de lucro,
chequea esto:



Inicio: Hoy, viernes, 13:05:06

Final: Hoy, viernes, 13:21:32

Preguntas: 10

Respuestas
válidas:

Puntuación:

Nota:

1 ¿Cuál cree que es la implementación óptima del siguiente algoritmo?
Elección única

```
int f(int n)
{
    int s = 0;
    for (int i = 0; i < n; ++i)
        s += i % 5 + 1;
    return s;
}
```

Usuario Profesores

- a) f(int):

```
xorl    %ecx, %ecx
xorl    %r8d, %r8d
movl    $5, %esi
.L3:
    cmpl    %edi, %ecx
    jge     .L1
    movl    %ecx, %eax
    incl    %ecx
    cltd
    idivl    %esi
    leal    1(%r8,%rdx), %r8d
    jmp     .L3
.L1:
    movl    %r8d, %eax
    ret
```
- b) f(int):

```
leal    (%rdi,%rdi,2), %eax
ret
```
- c) f(int):

tú puedes
ayudarnos a
llevar
WUOLAH
al siguiente
nivel
(o alguien que
conozcas)

```

    leal    0(,%rdi,4), %eax
    ret

```

d) ninguna otra respuesta es correcta

2

Elección única

¿Cuál de las optimizaciones siguientes reduce el tiempo de ejecución del código que se muestra a continuación? (M y N son múltiplos de dos).

```

for (i=0; i<M; i++)
    for (j=0; j<N; j++){
        if ((j%2)==0)
            c[j] += a[j][i]+b[j][i];
        else
            c[j] += a[j][i]-b[j][i];
    }

```

Usuario Profesores

a) for (i=0; i<M; i+=2)
 for (j=0; j<N; j+=2){
 c[j] += a[j][i]+b[j][i];
 c[j+1] += a[j+1][i+1]-b[j+1][i+1];
 }

b) for (i=0; i<M; i++)
 for (j=0; j<N; j+=2){
 c[j] += a[j][i]+b[j][i];
 c[j+1] += a[j+1][i]-b[j+1][i];
 }

c) for (i=0; i<M; i++)
 for (j=0; j<N; j++){
 c[j] += a[j][i]+b[j][i];
 c[j+1] += a[j+1][i]-b[j+1][i];
 }

d) for (j=0; j<M; j++)
 for (i=0; i<N; i++){
 if ((i%2)==0)
 c[i] += a[i][i]+b[i][j];
 else
 c[i] += a[i][j]-b[i][j];
 }

3

Elección única

¿Cuál de los siguientes códigos es computacionalmente más eficiente?

Usuario Profesores

a) x = w & 7;
 y = x * x;
 z = (y << 5)+y;
 for (i = h = 0 ; i < MAX ; i++) {
 h += 14;
 }


b) x = w % 8;
 y = x * x;
 z = (y << 5)+y;
 for (i = 0 ; i < MAX ; i++) {
 h = 14 * i;
 }

c) x = w % 8;
 y = pow(x, 2.0);
 z = y * 33;
 for (i = 0 ; i < MAX ; i++) {
 h = 14 * i;
 }

```

for (i = 0 ; i < MAX ; i++) {
    h = 14 * i;
}

```

 d) `x = w & 7;`
`y = pow (x, 2,0);`
`z = (y << 5)+y;`
`for (i = h = 0 ; i < MAX ; i++) {`
 `h += 14;`
`}`


4


Elección única


¿Qué código cree que calculará de forma correcta y en menor tiempo el producto de dos matrices en un sistema multiprocesador? Suponga matrices cuadradas, c inicializada a cero y N muy grande.


```
int a[N][N], b[N][N], c[N][N];
```

Usuario Profesores

 a) `for (int i=0; i<N; ++i)`
`#pragma omp parallel for`
`for (int j=0; j<N; ++j)`
`for (int k=0; k<N; ++k)`
 `c[i][j] += a[i][k] * b[k][j];`

 b) `for (int i=0; i<N; ++i)`
`#pragma omp parallel for`
`for (int j=0; j<N; ++j)`
`for (int k=0; k<N; ++k)`
`#pragma omp critical`
 `c[i][j] += a[i][k] * b[k][j];`

•  c) `for (int i=0; i<N; ++i)`
`#pragma omp parallel for`
`for (int j=0; j<N; ++j)`
`for (int k=0; k<N; ++k)`
`#pragma omp atomic`
 `c[i][j] += a[i][k] * b[k][j];`

 d) `for (int i=0; i<N; ++i)`
`for (int j=0; j<N; ++j)`
`for (int k=0; k<N; ++k)`
 `c[i][j] += a[i][k] * b[k][j];`

5


Elección única


¿Cómo cree que implementará el compilador una función que multiplica un entero por 11 al compilar con optimización máxima (-O3)?


```
int function_f(int x)
{
    return x * 11;
}
```

Usuario Profesores

 a) ninguna otra respuesta es correcta

 b) `0x401116 <+0>: imul $0xb,%edi,%eax`
`0x401119 <+3>: retq`

 c) `0x401120 <+0>: lea (%rdi,%rdi,4),%eax`
`0x401123 <+3>: lea (%rdi,%rax,2),%eax`
`0x401126 <+6>: retq`

•  d) `0x401106 <+0>: push %rbp`
`0x401107 <+1>: mov %rsp,%rbp`
`0x40110a <+4>: mov %edi,-0x4(%rbp)`
`0x40110d <+7>: mov -0x4(%rbp),%edx`
`0x401110 <+10>: mov %edx,%eax`

quieres trabajar
en Wuolah??

TE BUSCAMOS

28/5/2021

SWAD: plataforma de apoyo a la docencia / UGR /

```
0x401112 <+12>: shl    $0x2,%eax
0x401115 <+15>: add     %edx,%eax
0x401117 <+17>: add     %eax,%eax
0x401119 <+19>: add     %edx,%eax
0x40111b <+21>: pop     %rbp
0x40111c <+22>: retq
```

6

¿Cuál de las siguientes afirmaciones es correcta?

Elección única

Usuario Profesores

- ☐ a) hay optimizaciones que son aplicables a cualquier procesador
- ☐ b) el proceso de optimización se debe realizar siempre al final del desarrollo de la aplicación
- ☐ c) ninguna otra respuesta es correcta
- ☐ d) la optimización de código siempre debe realizarse en lenguaje ensamblador

sin ánimo
de lucro,
chequea esto:



7

¿Cuál de las siguientes versiones de una función que multiplica un entero por 6 cree que se obtendrá al compilar con optimización en espacio (-Os)?

Elección única

```
int f(int x)
```

```
{
    return x * 6;
}
```

Usuario Profesores

- ☐ a) 0x401116 <+0>: imul \$0x6,%edi,%eax
0x401119 <+3>: retq
- ☐ b) ninguna otra respuesta es correcta
- ☐ c) 0x401106 <+0>: lea (%rdi,%rdi,2),%eax
0x401109 <+3>: add %eax,%eax
0x40110b <+5>: retq
- ☐ d) 0x401106 <+0>: push %rbp
0x401107 <+1>: mov %rsp,%rbp
0x40110a <+4>: mov %edi,-0x4(%rbp)
0x40110d <+7>: mov -0x4(%rbp),%edx
0x401110 <+10>: mov %edx,%eax
0x401112 <+12>: add %eax,%eax
0x401114 <+14>: add %edx,%eax
0x401116 <+16>: add %eax,%eax
0x401118 <+18>: pop %rbp
0x401119 <+19>: retq

tú puedes
ayudarnos a
llevar
WUOLAH
al siguiente
nivel
(o alguien que
conozcas)

8

Escoja la mejor forma de calcular el valor de la sumatoria de la siguiente estructura:

Elección única

```
const int N = 1000;
struct S { int a[N], b[N]; } s
int sum = 0;
```

Usuario Profesores

- ☐ a) for (int i = 0; i < N; i += 2)
{
 sum += s.a[i] + s.a[i + 1];
 sum += s.b[i] + s.b[i + 1];
}
- ☐ b) for (int i = 0; i < N; ++i)
{
 sum += s.a[i];
}

WUOLAH

```

        sum += s.a[i];
        sum += s.b[i];
    }
    c) for (int i = 0; i < N; ++i)
        sum += s.a[i];
        for (int i = 0; i < N; ++i)
            sum += s.b[i];
    • d) for (int i = 0; i < N; i += 2)
        sum += s.a[i] + s.a[i + 1];
        for (int i = 0; i < N; i += 2)
            sum += s.b[i] + s.b[i + 1];

```

9

Elección única

Indique qué opción se ejecutará más rápido dados

```
const int n = 1000000;
```

```
int a[n], b[n];
```

Usuario Profesores

- ☐ a) for (i=0 ; i<n ; ++i) {
 *p += a[i]*b[i];
 }
- ☐ b) for (i=0 ; i<n ; i++) {
 *p = *p + a[i]*b[i];
 }
- ☒ c) int tmp0=0, tmp1=0, tmp2=0, tmp3=0;
 for (i=0 ; i<n ; i+=4) {
 tmp0 += a[i]*b[i];
 tmp1 += a[i+1]*b[i+1];
 tmp2 += a[i+2]*b[i+2];
 tmp3 += a[i+3]*b[i+3];
 }
 *p = tmp0 + tmp1 + tmp2 + tmp3;
- ☐ d) for (i=0 ; i<n ; i+=4) {
 *p += a[i]*b[i];
 *p += a[i+1]*b[i+1];
 *p += a[i+2]*b[i+2];
 *p += a[i+3]*b[i+3];
 }

10

Elección única

Dado el siguiente código y suponiendo el vector v inicializado, ¿qué opción es verdadera?

```

for (int i = 0; i < 1000; ++i)
{
    if ((v[i] % 3) == 0)
        foo(v[i]);
    else
        switch((v[i] % 3))
        {
            case 1: foo(v[i] + 2); break;
            case 2: foo(v[i] + 1); break;
        }
}

```

Usuario Profesores

- ☐ a) la ejecución finaliza antes si v no contiene ningún múltiplo de 3
- ☒ b) la ejecución finaliza antes si v contiene muchos múltiplos de 3
- ☐ c) los valores contenidos en v no afectan a la velocidad de

- c) los valores contenidos en V no afectan a la velocidad de ejecución
 - d) sólo el desenrollado de bucle puede servir para optimizar el código
-