

Student:

Collegekaartnummer:

Tentamen Computersystemen

baiCOSY06 2e jaar bachelor AI, 2e semester 27 februari 2012

vraag 1

Deze vraag gaat over het delen van *file descriptors* tussen twee processen die m.b.v. een *fork* zijn gecreeerd. Stel je hebt de *textfile* `./file1.txt` met de volgende inhoud:

```
aabbccdd
```

Deze *file* gaat gemanipuleerd worden door twee programmas. De volgende *source-code* is resp. gecompileerd tot `./program1` en `./program2`:

```
/*
 * Program1
 */
#include <unistd.h>
#include <fcntl.h>

int main()
{
    int pid, fd_x, fd_y, fd_z;
    char buf[8];

    fd_x = open("file1.txt", O_RDWR);
    fd_y = open("file1.txt", O_RDWR);
    fd_z = open("file1.txt", O_RDWR);

    read(fd_x, buf, 2);
    read(fd_y, buf+2, 4);

    if ((pid = fork()) == 0) {
        dup2(fd_x, STDOUT_FILENO);
        dup2(fd_y, STDIN_FILENO);
        execl("program2", "program2", NULL);
    }

    wait(NULL);

    read(fd_y, buf+6, 2);

    write(fd_z, buf+6, 2);
    write(fd_x, buf+4, 2);
    write(fd_x, buf+2, 2);

    close(fd_x);
    close(fd_y);
    close(fd_z);
}
```

Student:

Collegekaartnummer:

```
/*
 * Program2
 */
#include <unistd.h>
#include <fcntl.h>

int main()
{
    char buf[2];

    read(STDIN_FILENO, buf, 2);
    write(STDOUT_FILENO, buf, 2);
}
```

Geef aan op welke manier de inhoud van `file1.txt` door beiden programmas gemanuueerd wordt. Wat is de inhoud van `file1.txt` nadat `./program1` is uitgevoerd? U mag aannemen dat de *reads* and *writes* niet gecached zijn.

vraag 2

Deze opgave gaat over het optimaliseren van een functie op snelheid voor een Intel Pentium III. De functionele eenheden in een Pentium III hebben de volgende eigenschappen:

Operation	Latency	Issue Time
Integer Add	1	1
Integer Multiply	4	1
Integer Divide	36	36
Floating Point Add	3	1
Floating Point Multiply	5	2
Floating Point Divide	38	38
Load or Store (Cache Hit)	1	1

Bestudeer de volgende functies:

Loop 1	Loop 2
<pre>int loop1(int *a, int x, int n) { int y = x*x; int i; for (i = 0; i < n; i++) x = y * a[i]; return x*y; }</pre>	<pre>int loop2(int *a, int x, int n) { int y = x*x; int i; for (i = 0; i < n; i++) x = x * a[i]; return x*y; }</pre>

Student:

Collegekaartnummer:

Wanneer we deze code compileren met GCC, krijgen we de volgende code voor de *for-loop*:

Loop 1	Loop 2
<pre>.L21: movl %ecx,%eax imull (%esi,%edx,4),%eax incl %edx cmpl %ebx,%edx jl .L21</pre>	<pre>.L27: imull (%esi,%edx,4),%eax incl %edx cmpl %ebx,%edx jl .L27</pre>

(op bladzijde 205, 212, 220 en 224 staat een tabel met de betekenis van de IA32 instructies)

Als we deze code executeren op een Pentium III, meten we dat er voor `loop1` 3.0 *cycles* per iteratie nodig zijn, terwijl voor `loop2` 4.0 nodig zijn.

- Leg uit hoe het mogelijk is dat de code van `loop1` sneller is dan de code voor `loop2`, ondanks het feit dat voor deze code een *assembly* instructie extra nodig is.
- Als we de optie `-funroll-loops` meegeven tijdens het compileren, dan probeert de compiler een viervoudige *loop unrolling* toe te passen. Dit verhoogt de efficiency van `loop1`. Leg dit uit.
- Loop unrolling* heeft echter geen effect op de CPE van `loop2`. Leg dit uit.

vraag 3

- Waarom zijn RISC instructies makkelijker te implementeren in een pipeline dan CISC instructies?
- Is het zinvol de data en de instructies in verschillende caches onder te brengen? Verklaar je antwoord.
- Leg het verschil tussen *Big Endian* en *Little Endian* uit.

vraag 4

Je bent begonnen bij een bedrijf dat een aantal procedures gebruikt om 4 *signed* bytes op te slaan in een simpele datastructuur `packed_t` (32 bits *unsigned*). De bytes in de datastructuur `packed_t` zijn genummerd van 0 (*least significant*) tot 3 (*most significant*). Je krijgt de opdracht om een functie `xbyte` te schrijven voor een machine met *two's complement arithmetic* en *arithmetic right shifts*. De functie dient gebruik maken volgende definities:

```
/* Declaration of a data type where 4 bytes are packed into an unsigned */
typedef unsigned packet_t;
```

```
/* Extract byte from word. Return as signed integer */
int xbyte(packet_t word, int bytenum);
```

Student:

Collegekaartnummer:

De functie `xbyte` dient dus de desbetreffende byte uit de datastructuur halen en te converteren naar een *signed integer* met behulp van *sign extension*. Je voorganger (die ontslagen is vanwege complete incompetentie) heeft de volgende code geschreven:

```
/* Failed attempt to implement function */
int xbyte(packed_t word, int bytenum)
{
    return (word >> (bytenum << 3)) & 0xFF;
}
```

- a. Wat is er fout aan deze code?
- b. Geef een correcte implementatie van deze functie. Gebruik hierbij alleen maar *left en right shifts*, en eventueel een *subtraction*.

Succes!