Saarland University
Department 6.2 - Computer Science
Prof. Dr. W. J. Paul
Dr. Mikhail Kovalev, Jonas Oberhauser

System Architecture (block course) - SS13

Exercise Sheet 11 (due: 16.09.13) - 0 points, 23 bonus points

---

Organizational notes:

- This is the last exercise sheet. It doesn't have regular points, but only bonus points and the solutions to it will not be presented in the tutorial. If you have any questions regarding exercises from this exercise sheet, you can ask them during the tutorial on Monday or during the exam preparation sessions on Monday and on Tuesday.
- You can submit your solutions on Monday during the tutorial or you can bring it to Room 319 or Room 314 in E1.3 anytime before the tutorial.
- The last tutorial will be given next Monday.
- On Monday at 3.30pm (directly after the tutorial) and on Tuesday at 10:00am we will have exam preparation sessions hold by Prof. Paul in HS003 E1.3. We might also have another such session on Wednesday 10:00am.

Bonus Exercise 1:

(1 point (bonus)) Explain why $cvm.cp \in [0:p]$ but $cvm.vm \in [1:p] \to \text{ISA} - \text{sp}$.

Bonus Exercise 2:

(2 points (bonus)) Give, for $u \in [1:p]$, a formal definition of $cvm.pto(u)$ and $cvm.ptl(u)$.

Bonus Exercise 3:

Recall the process control blocks PCB and page table arrays PTA.

(a) (1 point (bonus)) What is stored in a single PCB?

(b) (1 point (bonus)) Give C0 type and variable declarations such that the following statements are true (p is a constant denoting the number of user processes):

$$\text{rfile} = \text{uint}[32]$$

$$\text{PCBE} = \{\text{rfile}\, gpr; \text{rfile}\, spr\}$$

$$\text{PCBT} = \text{PCBE}[p]$$

$$vtype(gm.PCB, c^0) = \text{PCBT}$$

(c) (1 point (bonus)) What is stored in the page table array?

(d) (1 point (bonus)) Give C0 type and variable declarations such that the following statements are true (k is a constant denoting the maximum number of pages):

$$\text{PTAT} = \text{uint}[k]$$

$$vtype(gm.PTA, c^0) = \text{PTAT}$$

(e) (1 point (bonus)) Extend these declarations by a function which takes a valid virtual address and a user and translates this virtual address into the physical address using the corresponding page table entry. Use the following framework:

Saarland University
Department 6.2 - Computer Science
Prof. Dr. W. J. Paul
Dr. Mikhail Kovalev, Jonas Oberhauser

System Architecture (block course) - SS13

Exercise Sheet 11 (due: 16.09.13) - 0 points, 23 bonus points

```
uint va2pa(uint va, uint u)
{
    uint px;
    uint bx;
    uint ptea;
    uint pte;
    uint ppx;
    uing pa;
    px   =                    ;
    bx   =                    ;
    ptea =                    ;
    pte  =                    ;
    ppx  =                    ;
    pa   =                    ;
    return pa;
}
```

Bonus Exercise 4:

(1 point (bonus)) Explain why we need compiler correctness to argue about C0 with inline assembler.

Bonus Exercise 5:

Recall the new component $c.rn$ that replaces $c.clr$.

(a) (2 points (bonus)) Explain why we replaced the component $c.clr$ with $c.rn \in [1 : c.rd] \to A$. Give the formal definition of $\delta_{C0}(c).rn$ in the case where $c.pr[1]$ is a function call.

(b) (1 point (bonus)) Explain why we can use $cvm'.c.rn = cvm.c.prn[1]$ in the case where a user process is issuing a system call.

Bonus Exercise 6:

(1 point (bonus)) Explain how we overloaded the function va and why we overloaded it.

Bonus Exercise 7:

(1 point (bonus)) Explain the semantics of the valid bit and how it is defined by the following bonuspart of the simulation relation between a machine running on its own and its implementation in the CVM kernel using virtualization:

$$cvm.vm(u).m(va) = \begin{cases} d.m(pma(d,u,va)) & v(d,u,va) = 1 \\ d.sm(sma(d,u,va)) & \text{o.w.} \end{cases}$$

Bonus Exercise 8:

(3 points (bonus)) What are the definitions of $B(cvm, d)$, $consis(k, d)$ and $kconsis(cvm, k)$.

Bonus Exercise 9:

Recall the definition of $cconf(d) := \{c | consis(c, d)\}$.

(a) (1 point (bonus)) Explain why $|cconf(d)| \neq 1$.

(b) (2 points (bonus)) Prove that if $c, c' \in cconf(d) \wedge \neg onheap(va(e, c))$ then $\neg onheap(va(e, c')) \wedge va(e, c) = va(e, c')$.

Saarland University
Department 6.2 - Computer Science
Prof. Dr. W. J. Paul
Dr. Mikhail Kovalev, Jonas Oberhauser

System Architecture (block course) - SS13

Exercise Sheet 11 (due: 16.09.13) - 0 points, 23 bonus points

---

Bonus Exercise 10:

(2 points (bonus)) Give, for $t \in TN$ and $v \in \mathbb{B}^{size(t)}$, a definition of a function $decode(v,t) \in ra(t)$ such that $enc(decode(v,t),t) = v$. Prove the correctness of your definition.

Bonus Exercise 11:

(1 point (bonus)) Recall that for defining the semantics of an inline assembler statement, we have to consider a MIPS configuration d and reconstruct a C0 configuration c such that $consis(c,d)$. Assume we have already reconstructed $c.rd$, $c.st(i)$ and $c.m(ST(i,c))$ for $i \in [0 : c.rd]$, as well as $c.m(gm)$. Reconstruct $c.rds(i)$ for $i \in [1 : c.rd]$ if $onstack(c.rds(i))$ or $ingm(c.rds(i))$.

Bonus Exercise 12:

(1 point (bonus)) Complete the statement of the following theorem:

$$\forall (x.ba, x.t) \in oris(j).(\exists j \in SV(c).x.ba = ba(y,c) \wedge x.t = vtype(y,c)) \\ \vee \ldots$$

Bonus Exercise 13:

(1 point (bonus)) Recall the definition of $oris(0)$:

$$\begin{aligned} oris(0) = &\{(sbase, \$gm)\} \\ &\cup \{(ba(ST(i,c),c), \$c.st(i)) \mid i \in [0 : c.rd]\} \\ &\cup \{(ba(ST(i,c),c) -_{32} 4_{32}, ft(c.st(i)).t*) \mid i \in [0 : c.rd]\} \end{aligned}$$

Change this definition to include the variables pointed at by $c.rds(i)$ rather than $\{(ba(ST(i,c),c) -_{32} 4_{32}, ft(c.st(i)).t*) \mid i \in [0 : c.rd]\}$.

Bonus Exercise 14:

(1 point (bonus)) Explain why it is "over" once we know that $\phi \in oris(c) \to oris(c')$ is bijective.