

INFORMATICA INSTITUUT, FACULTEIT WISKUNDE EN INFORMATICA, UU.
IN ELEKTRONISCHE VORM BESCHIKBAAR GEMAAKT DOOR DE \mathcal{TC} VAN A-Eskwadraat.
HET COLLEGE INFONW WERD IN 2003/2004 GEGEVEN DOOR PIET VAN OOSTRUM.

Netwerken (INFONW)

10 december 2003

N.B. De \mathcal{TC} heeft alleen de beschikking over de open vragen.

Opgave 1

Zie de bijgevoegde lijst met SSM instructies op de volgende pagina.

- a) Schrijf de SSM code voor de methode `abs` die de absolute waarde van een parameter x uitrekent en teruggeeft als resultaat. D.w.z. x als $x \geq 0$ en $-x$ als $x < 0$.
- b) Schrijf de SSM code voor de methode `main` die bij `abs` aanroep met als parameter het getal -5 , gevolgd door een aanroep met als parameter het getal 7 , en die beide resultaten met elkaar vermenigvuldigt.

Opgave 2

In de meeste operating systems wordt een onderscheid gemaakt tussen *user mode* en *system mode* van de CPU. Leg uit wat het verschil is en hoe het systeem van user mode in system mode komt en omgekeerd.

Opgave 3

Leg uit wat het verschil is tussen preemptive scheduling en non-preemptive scheduling.

Opgave 4

Hoe komt het dat in UNIX wel *links* in het filesysteem mogelijk zijn en op Windows 98 niet?

Opgave 5

Geef met schematische code aan (dit hoeft geen correct C ofzo te zijn) hoe I/O direction samenhangt met *fork* en *exec* in Unix.

Opgave 6

Bespreek belangrijke verschillen tussen IPC met anonieme pipes en met message passing. Geef duidelijk aan hoe deze beide mechanismes daarin verschillen en wat de voor- en nadelen ervan zijn.

Opgave 7

Welke waarden worden er bij RPC in de client stub gemarshalled? En welke geunmarshalled? En welke waarden worden in de server stub gemarshalled en welke geunmarshalled?

Overzicht SSM instructies

instructie	pop	push	operatie
add	2	1	$a + b$
sub	2	1	$a - b$
mul	2	1	$a \times b$
div	2	1	a/b
mod	2	1	$a\%b$
neg	1	1	$-b$
eq	2	1	$a == b$
ne	2	1	$a! = b$
lt	2	1	$a < b$
gt	2	1	$a > b$
le	2	1	$a \leq b$
ge	2	1	$a \geq b$
ldc X	0	1	load constante X
ldr R	0	1	load register R
lds n	0	1	load woord n t.o.v. SP
ldl n	0	1	load woord n t.o.v. MP
str R	1	0	store register R
sts n	1	0	store woord n t.o.v. SP
stl n	1	0	store woord n t.o.v. MP
brt	1	0	spring als b true is
brf	1	0	spring als b false is
bra	0	0	spring altijd
bsr	0	1	branch to subroutine
ret	1	0	keer terug na methodeaanroep
halt	0	0	stopt SSM
link n	0	$n + 1$	zet MP, reserveer lokale variabelen
unlink n	$n + 1$	0	omgekeerde van link n
ajs n	*	*	$SP = SP + n$