Name: Flori Kusari — Problem Sheet #12 — Date: 29/11/2023

12.1 (a) and (b) have been submitted as text files.

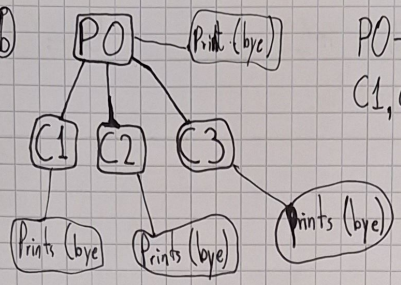
12.2 operating system processes

(a) Annotate the assembler instructions:

- `addi sp, sp, -16`: This subracts 16 from the stack pointer sp, effectively allocating 16 bytes on the stack for the current function stack frame.
- `sd ra, 8(sp)`: Saves the return address ra to location sp+8.
- `sd s0, 0(sp)`: Saves frame pointer s0 to memory location sp.
- `addi s0, sp, 16`: Adds 16 to the stack pointer sp and stors result in s0
- `jal fork`: Jumps to fork function and saves the return address in ra
- `bne a0, zero, done`: This brandes to the done label if a0 ≠ 0. (return value of the fork)
- `jal execvp`: Jumps to execvp function which replaces the current process image with a new one
- `la a0, bye`: Loads address of the "bye" string into a0.
- `jal puts`: Jumps to puts function, prints the a0 pointed string to the standard output.
- `mv a0, zero`: Moves the value 0 into a0.
- `ld ra, 8(sp)`: Loads the return address ra from sp+8.
- `ld s0, 0(sp)`: Loads the framer pointer s0 from memory location at sp
- `addi sp, sp, 16`: This instruction adds 16 to the stack pointer sp, deallocating the current function's stack frame.
- `ret`: Returns address stored in ra.

BRUNNEN

b)



P0 - initial parent process that starts the programm
C1, C2, C3 : Child processes created by the fork calls in P0.

c) The program prints "bye" four times due to three forks creating eight processes. The "execp" call replaces these processes with the "date" command, preventing them from reaching the "puts" call. The remaining four processes, including the parent and the fork's children reach the "puts" call and print "bye".

Problem 12.3: Pre- and post conditions

a) After lines: ① $Z = X \cdot 2$
② $Y = Z + 2 = X \cdot 2 + 2$
③ $X = Y \cdot 3 = ((X \cdot 2) + 2) \cdot 3)$

So the strongest post condition is $(((X \cdot 2) + 2) \cdot 3) = X$ and $((2 \cdot X) + 2) = Y$.

b) After lines: ① $X = X + 2$
② $Y = X \cdot 4 = (X + 2) \cdot 4$
③ $X = Y - 4 = (X + 2) \cdot 4) - 4$

Plug ing in the postcondition $((X > 10) \wedge (Y < 20))$ we can plugin $X = ((X+2) \cdot 4) - 4$ which gives us $\boxed{X > \frac{3}{2}}$ and we do the same for $Y < 20$ which gives us $\boxed{X < 3}$ so the weakest pre conditions are $\boxed{X > \frac{3}{2}}$ and $\boxed{X < 3}$ aka. $\boxed{1.5 < X < 3}$.

c) For this we must work backwards to find it:

Given the post condition $7 \leq Y < 25$.

Starting from the last line: $Y := Y - 2$ we need $9 \leq Y < 27$

Now we go to the conditional statements.

Else:

$Y := X + 6$ $(x >= 12)$

we need: $3 \leq X < 21$

IF-Completed:

$Y := 3 \cdot x - 9$ (when $x < 12$)

we need: $6 \leq x < 12$

Combine: $3 \leq X < 21$

Moving up we get the line: $X := 3 \cdot Y - 2$

To ensure: $3 \leq x < 21$

we need: $\boxed{5/3 \leq Y < 23/3}$

Thus the weakest precondition has been found to be $\boxed{5/3 \leq Y < 23/3}$

THE END

BRUNNEN