

Date: 11-10-21 :

Vfork()

- Vfork() system call also creates the child process like fork(), but blocks the parent until child termination.
- Behaviour for Vfork is undefined in the following cases:
 - (i) modifying any data other than a variable of type pid_t.
 - (ii) if there is a return in the function where vfork() called

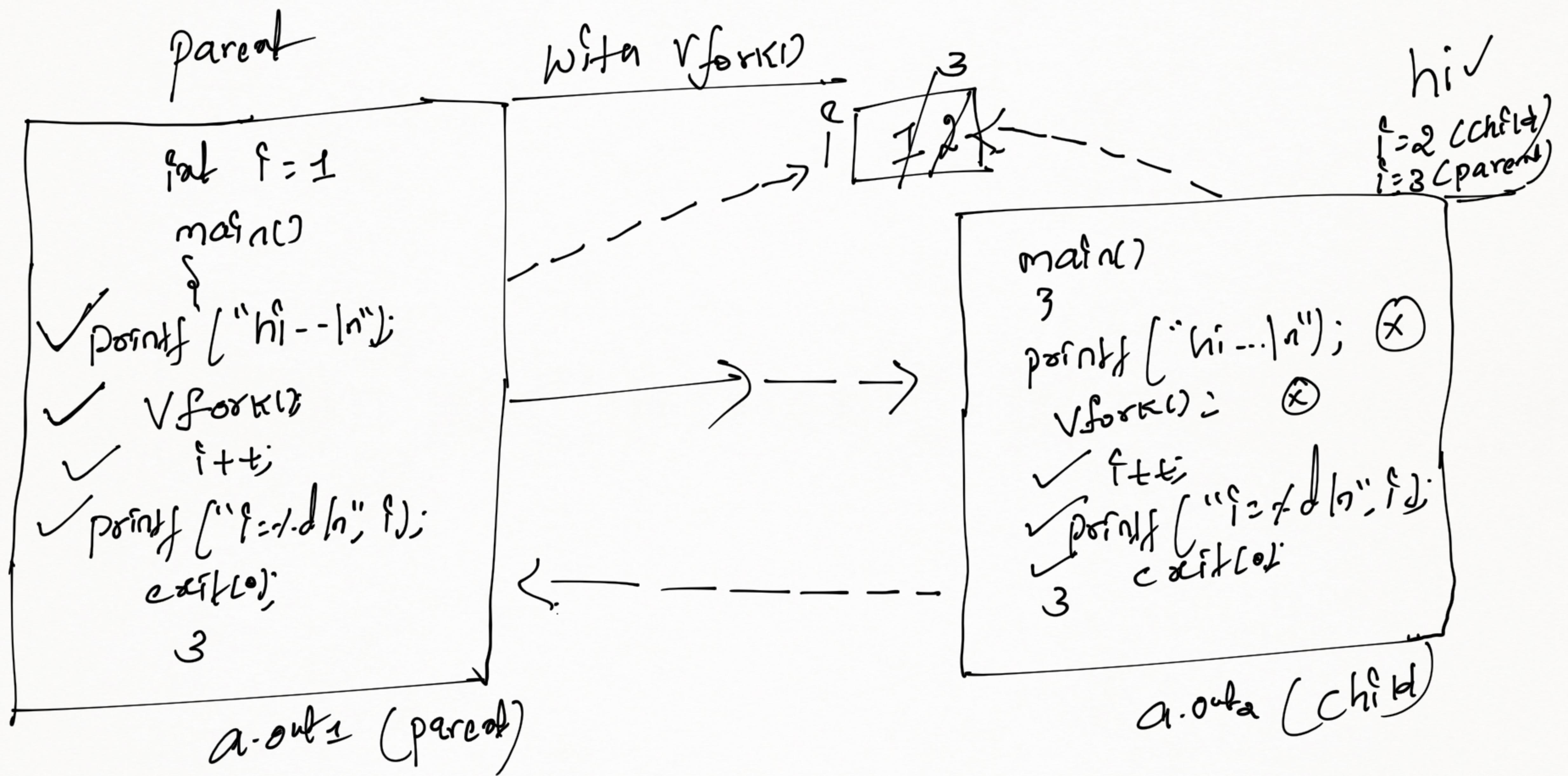
(C/C) if there is no exit/exit

✓ $i = 1$
main()
└
✓ printf("hi - ln")
✓ fork();
└
✓ printf(" i = - / . hi : ? ");
└ 3

(X) a. only parent



hi ✓ $i = 2$
i = 2
main()
└
printf("hi - ln"); (X)
— fork(); (X)
✓ i + ;
✓ printf(" i = - / . hi : ? ");
└ 3
(child)



fork()

① fork() creates the child process

② after creating child process
both parent and child will run
parallel. [Job concorenct]

vfork()

vfork() also creates child
process

after creating child process,
parent process blocks until
child termination. (In case
of vfork() child executed first)
here parallel execution not
possible

③ In case `fork()`, if any one of the process (parent) child) modifying the common data separate copy created (CoW)

→ In case of `Vfork()`, after child process creation if any one of the process-modifying common data no separate copy.
(Both parent & child will run under same address space).

④ modification for data by one process will not affect other process

→ modification for data by one process will affect other process.

- vfork() Limitations
- No Duplication of memory pages done for child,
Instead of that child share the parent's memory.
 - Execution of parent suspended until child has
terminated