A process forks several children with different behaviors

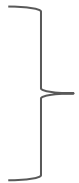
```
main ()
{
   cpid_1 = fork();
   if (cpid_1 == 0)
   { child_1's behavior defined here }

   cpid_2 = fork();
   if (cpid_2 == 0)
   { child_2''s behavior defined here }

   cpid_3 = fork();
   if (cpid_3 == 0)
   { child_3''s behavior defined here }


   doing_something_else();
   waitpid(cpid_1);
   waitpid(cpid_2);
   waitpid(cpid_3);

}
```



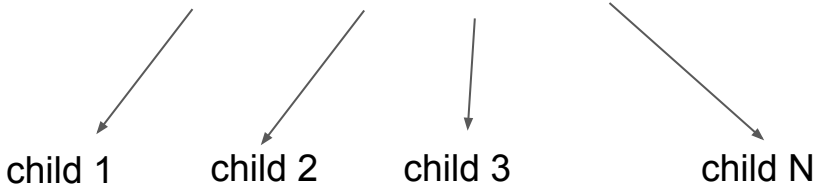
parent continues here

Auto IPC communication among children processes due to inheritance of attached shared memory segment

Children processes created by calls to fork inherit attached shared segments; they can detach the shared memory segments, if desired.

Step1: parent process allocates and attaches a shared segment

Step2: parent process forks a number of children processes



Children processes can communicate with each other due to inheritance of their parent's attached shared segment. They don't need explicit attachment.

Step3: parent process waits all children to exit, and then de-attaches the shared memory segment, and finally it deallocates the shared segment