

20 a) O que este código faz? É possível prever qual será a saída do programa?

O processo-pai cria 2 filhos, que farão coisas análogas (ler do ~~buffer~~ ^{pipe} e então escrever no pipe ("tomato" no filho 1 e "turnip" no 2)), ele então escreve no pipe, printa uma mensagem em 'stdout', espera um pouco, lê do pipe e escreve isso em 'stdout' junto de outra mensagem, e mata seus filhos. Não dá para prever a saída do programa (o que o pai lerá do pipe) pois isso depende de qual foi o último filho a ler-e-escrever no pipe, e ambos estão em loop infinito sem ~~estados~~ condições de controle.

b) O que acontece se omitirmos o comando **write** do pai?

Os três processos entrarão em deadlock, pois todos tentarão ler de uma pipe vazia e ficarão travados nesse ponto, pois não há ninguém para escrever na pipe.

c) O que acontece se omitirmos o comando **write** de um dos filhos?

~~Quando o processo pai chamar o 'write' de um dos filhos seja removido, podemos garantir a saída do programa, pois só haverá uma coisa sendo escrita na pipe (isso porém, apenas se o processo sem write não for chamado entre o pai e o filho com write)~~