

2) Basicamente os dois filhos do processo ficam "brigando" durante os 10 segundos que o pai ~~deixa~~ removendo o conteúdo do pipe (lê e não faz nada com o conteúdo) e escrevendo "tomato" para o filho 1 e "turnip" para o filho 2.

Ao final dos 10 segundos, quando o pai tomar a posse ele lê o que está no pipe e este conteúdo será o último filho que escrever (Supondo que durante estes 10 segundos os filhos foram escalonados, caso contrário ele iria ler "potato") e imprimir na tela. Em seguida, se os filhos foram escalonados, estes permanecerão bloqueados pois o pipe está cheio, se não, o pai mata os filhos e libera os ~~lombos~~ deles e encerra.

Importante lembrar, que ao iniciar o programa, se os filhos forem escalonados primeiro estes permanecerão bloqueados pois realmente é bloqueante caso o pipe esteja cheio. São desbloqueados quando o pai fizer o write e em seguida escalonados.



b) Exatamente não, pois depende da ordem de escalonamento. Mas pode ser as seguintes possibilidades:

Parent: I wrote a potato!

Parent: I got back a tomato!

OU

Parent: I wrote a potato!

Parent: I got back a turnip!

OU (Caso extremamente raro)

Parent: I wrote a potato!

Parent: I got back a potato!

c) 2 situações:

1) Retirar write de um filho e este nunca ser escalonado: O programa funciona normalmente e o filho que tem o write terá seu fruto impresso pelo pai.

2) Retirar write de um filho mas ambos ou apenas este for(em) escalonado(s): Deadlock. Em algum momento, o filho com o write ficará bloqueado no read esperando conteúdo do pipe e o pai também. O filho sem o write também.

Caso extremo: Filhos nunca escalonados: Pai imprime potato no final e programa funciona normalmente.