Turgut Alp Edis
21702587

CS342 - HW 3

1) Alloc Need

P1   x   10-x

P2   y   12-y

Available : $15-x-y$

$P_1 \rightarrow P_2$ or $P_2 \rightarrow P_1$

For $P_1 \rightarrow P_2$:

$10-x \leq 15-x-y$

$y \leq 5$

For $P_2 \rightarrow P_1$:

$12-y \leq 15-x-y$

$x \leq 3$

Therefore $y \leq 5$ or $x \leq 3$ makes the process thread safe.

So, there are 94 safe states.

2)

Yes, because after allocations, available: 3, 3, 0

P.4 can satisfy ⟶ avaible: 4, 3, 1

P1 can satisfy ⟶ available: 5, 3, 1

then, none of the P2, P3 and P5 cannot satisfy the avaible.

4)

a) $2^{36}/16 \, KB = 2^{36}/2^{14} = 2^{22}$ entries

$2^{22} \cdot 2^3$ (8 byte long entry) $= 2^{25} = 32 \, MB$

b) Second level table => 2^{11} entries => $2^{11} \cdot 2^{14} = 2^{25} = 32 \, MB$ can be mapped from virtual memory to physical memory.

$128/32 + 64/32 + 32/32 = 7$ second level pages

1 is total top level page

A second level table => $2^{11} \cdot 2^3 = 2^{14}$

A top level table => $2^{11} \cdot 2^3 = 2^{14}$ } required mem. spaces

Hence, $7 \cdot 2^{14} + 1 \cdot 2^{14} = 128 \, KB$

c) Inverted page $\Rightarrow$ $4GB/16KB = 2^{32}/2^{14} = 2^{18}$ entries
8 byte (entry size)

$2^{18} \cdot 2^3 = 2^{21} = 2MB \Rightarrow$ size of inverted page table

3)

2	5	3	5	6	3	8	5	4	1	2	4	8	5	3	8	4	3	1	5

5)

a) 12 direct pointers $\Rightarrow 12 \cdot 4KB = 48KB$

1 single indirect pointer $\Rightarrow 4KB \cdot (4KB/4) = 4MB$

1 double indirect pointer $\Rightarrow 1KB \cdot 1KB \cdot 4KB = 4GB$

1 triple indirect pointer $\Rightarrow 1KB \cdot 1KB \cdot 1KB \cdot 4KB = 4TB$

The max file size is $4TB + 4GB + 4MB + 48KB$

6) No solution

7)
   a) Parity blocks are distributed evenly so that data is distributed among all disks,
     Corruption can be found with XOR operation
   b) No solution

8) RAID4 $\longrightarrow$ 9 disks $MTTDL = \dfrac{MTTF^2}{N \times MTTR} = \dfrac{308641}{432} \sim \boxed{714}$
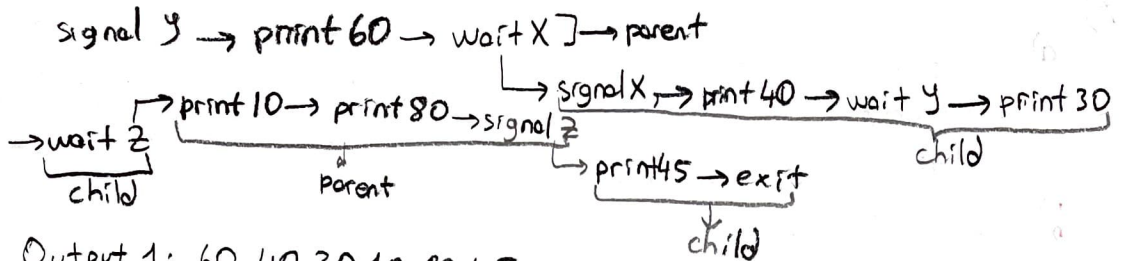
   MTTF = 50000 hours

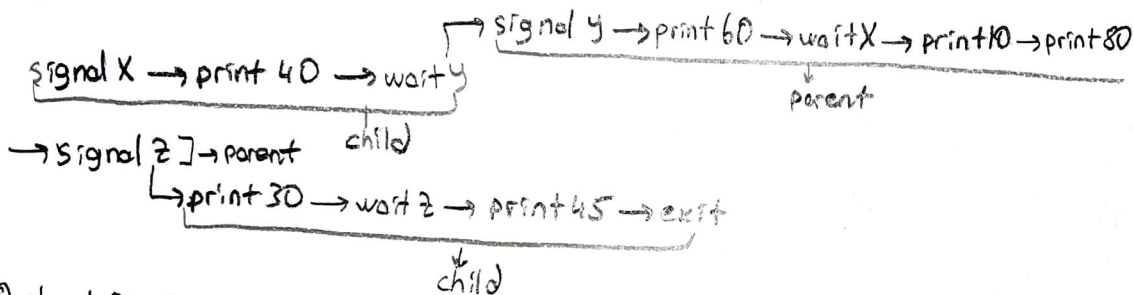   MTTR = 48 hours

   MTTF of 9 $\Rightarrow$ 50000/9 = 555,55

9)

11) Output depends on OS, so there are two different scenarios.

Scenario 1: Parent executes first

signal y → print 60 → wait X] → parent
  └→ signal X → print 40 → wait y → print 30 *(child)*

→ print 10 → print 80 → signal z *(parent)*

→ wait z *(child)*

  └→ print 45 → exit *(child)*

Output 1: 60 40 30 10 80 45

Scenario 2: Child executes first

signal y → print 60 → wait X → print 10 → print 80 *(parent)*

signal X → print 40 → wait y *(child)*

→ signal z] → parent

  └→ print 30 → wait z → print 45 → exit *(child)*

Output 2: 40 60 10 80 30 45