

OS 2020 Problem Sheet #5

Problem 5.1: safe states

(3 points)

A system has $n = 4$ processes, $m = 5$ resource types, and the number of resources for each resource type is given by $t = (6, 15, 8, 10, 9)$. The system is in the following state:

$$M = \begin{bmatrix} 3 & 5 & 8 & 10 & 1 \\ 2 & 5 & 3 & 3 & 2 \\ 4 & 12 & 4 & 9 & 2 \\ 6 & 1 & 4 & 5 & 5 \end{bmatrix} \quad A = \begin{bmatrix} 0 & 2 & 1 & 1 & 1 \\ 0 & 5 & 3 & 1 & 1 \\ 0 & 7 & 1 & 2 & 1 \\ 3 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Is the system in a safe state? Provide a calculation to justify your answer.

Problem 5.2: deadlock detection

(1+1 = 2 points)

A system has $n = 3$ processes, $m = 4$ resource types, and the number of resources for each resource type is given by $t = (1, 2, 1, 3)$. The system is in the following state:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad N = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

- Draw the corresponding resource allocation graph.
- Is the system deadlocked? Provide a calculation to justify your answer.

Problem 5.3: scheduling

(1+1 = 2 points)

Five processes arrived in the order $P_4 \prec P_1 \prec P_5 \prec P_3 \prec P_2$ and they are all ready at time $t = 0$. Process P_1 needs 2 time units, P_2 needs 3 time units, P_3 needs 6 time units, P_4 needs 10 time units, and P_5 needs 19 time units. There is only one CPU in the system.

- Draw the resulting schedule for the scheduling strategies first-come first-served (FCFS), longest processing time first (LPTF), shortest job first (SJF) and round robin (RR) with a time slice of 2 time units.
- For each schedule, calculate the average completion time \bar{e} .

Problem 5.4: linking

(2+1 = 3 points)

The following C source files are compiled separately into object files and afterwards linked with other object files into an executable.

```

1  /* a.c */
2
3  #include <stdio.h>
4
5  extern int x;
6  int y;
7
8  static void f()
9  {
10     printf("a.c: f()\n");
11 }
12
13 void g()
14 {
15     printf("a.c: g()\n");
16     f();
17 }

```

```

1  /* b.c */
2
3  #include <stdio.h>
4
5  extern void g();
6
7  int x = 1;
8  static double y = 1;
9
10 void f()
11 {
12     printf("b.c: f()\n");
13     g();
14 }

```

a) Which symbols defined in the files a.c and b.c are

- internally defined symbols not accessible outside of the object file,
- references to externally defined symbols that must be resolved by the linker,
- weak linkable symbols defined in the object file, or
- strong linkable symbols defined in the object file?

Mark the corresponding cell in the following table (we ignore the printf symbol).

file	symbol	internal	external	weak symbol	strong symbol
a.c	x				
a.c	y				
a.c	f				
a.c	g				
b.c	x				
b.c	y				
b.c	f				
b.c	g				

b) What will be printed to the standard output by the following main() function? Explain.

```

1  /* main.c */
2
3  extern void f();
4
5  int main()
6  {
7     f();
8     return 0;
9  }

```