Operating Systems
Jacobs University Bremen
Dr. Jürgen Schönwälder

Module: CO-562
Date: 2022-10-13
Due: 2022-10-20

OS 2022 Problem Sheet #6

Problem 6.1: *scheduling strategies* (4+2 = 6 points)

A computer system with a single CPU has to execute $n = 6$ processes $A, \ldots, F$. The arrival times and the execution times of the processes are given by the following table.

process	arrival time	execution time
A	0	7
B	3	5
C	5	9
D	8	3
E	10	1
F	12	2

a) Draw the schedule for the scheduling strategies first-come first-served (FCFS), shortest processing time first (SPTF), longest processing time first (LPTF), and round robin (RR) with a time slice of 1 time unit. Assume that arrivals happen before a scheduling point and that new processes are added at the end of the run queue.

b) For each schedule, calculate the average turnaround time $\bar{t}$ and the average waiting time $\bar{w}$.

Problem 6.2: *linking* (2+1+1 = 4 points)

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

```
/* a.c */

#include <stdio.h>

extern int x;
int y;

static void f()
{
    static char z = 'Z';
    puts("a.c: f()");
}

void g()
{
    puts("a.c: g()");
    f();
}

void h()
{
    puts("a.c: h()");
    g();
}
```

```
/* b.c */

#include <stdio.h>

extern void h();

int x = 1;
static double y = 1;
static char z = 'A';

static void g()
{
    puts("b.c: g()");
    h();
}

void f()
{
    puts("b.c: f()");
    g();
}
```

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 `puts` symbol).

file	symbol	internal unlinkable symbol	reference of external symbol	weak linkable symbol	strong linkable symbol
a.c	x				
a.c	y				
a.c	f				
a.c	g				
a.c	h				
b.c	x				
b.c	y				
b.c	z				
b.c	f				
b.c	g				

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

```
/* main.c */

extern void f();

int main()
{
    f();
    return 0;
}
```

c) What is name mangling and why do programming languages like C++ use name mangling? Why do I sometimes need to use `extern "C" {}` in C++ header files?