PROJECT 2

Priority Scheduling

Real time systems

Course 2022/2023

Participants:

Name and Surname	NIA
COSMIN OCTAVIAN PETRE CARASTOIAN	100428943
JORGE MATARIN SERRANO	100429088
DIANA GALLARDO DE PEDRO	100406633

PART 1: 1-bit samples

Software module

Scheduler

The tasks are not independent because there is a shared variable. Therefore, we will use the priority inheritance method. All times are in milliseconds:

	T=D	C	P	LOCK
T1: PLAYBACK TASK	512	7	3 +	1
T2: RECEIVE COMMANDS	1000	1	2	1
T3: SHOW PLAYBACK STATE	2500	4	1 -	1

CPU usage: $U = 7/512 + 1/1000 + 4/2500 = 0.0163 < 1$

$B(T1) = 1 + 1 = 2$ $\qquad$ $B(T2) = 1$ $\qquad$ $B(T3) = 0$

$R(T1) = 7 + 2 + 0 = 9 \preccurlyeq 512 = D(T1)$

$$
\begin{aligned}
R(T2) &= 1 + 1 + 7 * \lceil R(T2)/512 \rceil \\
W0 &= 1 + 1 + 7 * \lceil 1/512 \rceil = 9 \\
W1 &= 1 + 1 + 7 * \lceil 9/512 \rceil = 9 \\
R(T2) &= 9 \preccurlyeq 1000 = D(T2)
\end{aligned}
$$

$$
\begin{aligned}
R(T3) &= 4 + 0 + 7 * \lceil R(T3)/512 \rceil + 1 * \lceil R(T3)/1000 \rceil \\
W0 &= 4 + 7 * \lceil 1/512 \rceil + 1 * \lceil 1/1000 \rceil = 12 \\
W1 &= 4 + 7 * \lceil 12/512 \rceil + 1 * \lceil 12/1000 \rceil = 12 \\
R(T3) &= 12 \preccurlyeq 2500 = D(T3)
\end{aligned}
$$

Hardware module

Scheduler

The tasks are not independent because there is a shared variable. Therefore, we will use the priority inheritance method. All times are in microseconds:

	T=D	C	P	LOCK
T1: PLAYBACK	250	28	2 +	28
T23 (2 & 3): BUTTON AND LED	1000	24	1 -	8

CPU usage: $U = 28/250 + 24/1000 = 0.136 < 1$

$B(T1) = 8$ $B(T23) = 0$

$R(T1) = 28 + 8 + 0 = 36 \leqslant 250 = D(T1)$

$$\begin{aligned}
R(T23) &= 24 + 0 + 28 * \lceil R(T23)/250 \rceil \\
W0 &= 24 + 0 + 28 * \lceil 1/250 \rceil = 52 \\
W1 &= 24 + 0 + 28 * \lceil 52/250 \rceil = 52 \\
R(T23) &= 52 \leqslant 1000 = D(T23)
\end{aligned}$$

PART 2: 8-bit samples

Software module

Scheduler

The tasks are not independent because there is a shared variable. Therefore, we will use the priority inheritance method. All times are in milliseconds:

	T=D	C	P	LOCK
T1: PLAYBACK TASK	64	7	3 +	1
T2: RECEIVE COMMANDS	1000	1	2	1
T3: SHOW PLAYBACK STATE	2500	4	1 -	1

CPU usage: $U = 7/64 + 1/1000 + 4/2500 = 0.112 < 1$

$B(T1) = 1 + 1 = 2$ $B(T2) = 1$ $B(T3) = 0$

$R(T1) = 7 + 2 + 0 = 9 \preccurlyeq 64 = D(T1)$

$R(T2) = 1 + 1 + 7 * \lceil R(T2)/64 \rceil$

$W0 = 1 + 1 + 7 * \lceil 1/64 \rceil = 9$

$W1 = 1 + 1 + 7 * \lceil 9/64 \rceil = 9$

$R(T2) = 9 \preccurlyeq 1000 = D(T2)$

$R(T3) = 4 + 0 + 7 * \lceil R(T3)/64 \rceil + 1 * \lceil R(T3)/1000 \rceil$

$W0 = 4 + 7 * \lceil 1/64 \rceil + 1 * \lceil 1/1000 \rceil = 12$

$W1 = 4 + 7 * \lceil 12/64 \rceil + 1 * \lceil 12/1000 \rceil = 12$

$R(T3) = 12 \preccurlyeq 2500 = D(T3)$

Hardware module

Scheduler

The tasks are not independent because there is a shared variable. Therefore, we will use the priority inheritance method. All times are in microseconds:

	T=D	C	P	LOCK
T1: PLAYBACK	250	28	2 +	28
T23 (2 & 3): BUTTON AND LED	1000	24	1 -	8

CPU usage: $U = 28/250 + 24/1000 = 0.136 < 1$

$B(T1) = 8 \qquad\qquad B(T23) = 0$

$R(T1) = 28 + 8 + 0 = 36 \preccurlyeq 250 = D(T1)$

$$
\begin{aligned}
R(T23) &= 24 + 0 + 28 * \lceil R(T23)/250 \rceil \\
W0 &= 24 + 0 + 28 * \lceil 1/250 \rceil = 52 \\
W1 &= 24 + 0 + 28 * \lceil 52/250 \rceil = 52 \\
R(T23) &= 52 \preccurlyeq 1000 = D(T23)
\end{aligned}
$$

Remarks

All compute times and lock times were estimated running the actual programs. Real compute times and lock times may be smaller than those used for priority scheduler design. Since these times depend on the hardware used, we should note that the hardware module was run on an Arduino Uno microcontroller, and the software module was run on QEMU on top of a Debian linux bare metal install.

We considered that the lock time of the playback task (task 1) in the hardware module for both parts was the same as the total compute time, since this task is essentially fully enclosed in a critical section. This is because this task is executed upon an interrupt, which can never be interrupted by another task, as we do not use other interrupts.

$OCR1A = Arduino\_frequency/(1/t\_interrupt) - 1 = 16000000/(1/0.00025) - 1 = 3999$