

- 1) Napište výsledné hodnoty následujících posunů – výsledek запиšte ve stejné soustavě (dvojková nebo šestnáctková) v jaké je zapsán levý operand operace posunu. Dodržte uvedenou přesnost operace:

8-bitový posun: 00101110 SHL 5
 8-bitový posun: 00101110 SHL 16
 8-bitový posun: 0x1C SHL 4
 32-bitový posun: 0x00017000 SHL 3
 32-bitový posun: 0x84ACFF09 SHL 8
 32-bitový posun: 0x84ACFF09 SHL 12

8-bitový posun: 10101110 SHR 5
 8-bitový posun: 10101110 SHR 7
 8-bitový posun: 00101111 SHR 7
 32-bitový posun: 0x84ACFF09 SHR 8
 32-bitový posun: 0x84ACFF09 SHR 12

- 2) Napište výsledné hodnoty následujících rotací – výsledek запиšte ve stejné soustavě (dvojková nebo šestnáctková) v jaké je zapsán levý operand operace posunu. Dodržte uvedenou přesnost operace:

8-bitová rotace: 00101110 ROL 5
 8-bitová rotace: 00101110 ROL 8
 8-bitová rotace: 0x1C ROL 4
 32-bitová rotace: 0x84ACFF09 ROL 8

8-bitová rotace: 10101110 ROR 3
 8-bitová rotace: 10101110 ROR 7
 32-bitová rotace: 0x84ACFF09 ROR 12

- 3) Pro následující hodnoty napište jejich reprezentaci jako znaménkové číslo (a) s explicitním znaménkovým bitem, (b) v jedničkovém doplňku, (c) ve dvojkovém doplňku. Po každou hodnotu napište všechny 3 varianty její reprezentace, a výsledek vždy запиšte jako 8-bitovou hodnotu ve dvojkové soustavě:

10
 0
 -3
 -5
 -124

- 4) Pro následující hodnoty N napište rozsah reprezentovatelných čísel v N-bitové přesnosti pro každou z reprezentací znaménkových čísel (a) s explicitním znaménkovým bitem, (b) v jedničkovém doplňku, (c) ve dvojkovém doplňku, a (d) pro bezznaménková čísla:

8
 7
 9
 12
 16
 24 (stačí přibližně)
 32 (stačí přibližně)

- 5) Srovnajte výhody a nevýhody reprezentace znaménkových čísel (a) s explicitním znaménkovým bitem, (b) v jedničkovém doplňku, (c) ve dvojkovém doplňku.

- 6) Pro následující hodnoty napište kolik bitů si bude Python doopravdy pamatovat pro proměnnou typu `int`, a kolik pro takovou proměnnou vrátí funkce `bit_length()`:

10
 255
 256
 -3
 -5
 -124

- 7) Jak byste v Pythonu s využitím modulu numpy zapsali, pokud chcete do proměnné `cislo` uložit níže uvedené hodnoty v uvedených reprezentacích:
- 8, jako 8-bitové bezznaménkové číslo
 - 256, jako 16-bitové bezznaménkové číslo
 - 9, jako 32-bitové znaménkové číslo
 - 15, jako 8-bitové znaménkové číslo
- 8) Předpokládejte následující fragmenty Python programů využívajících modul numpy. Pokud bychom po každém z těchto fragmentů provedli v programu příkaz `print(cislo, type(cislo))`, tak co by v uvedených příkladech vypsal?
- a): `cislo = 5`
 - b): `cislo = 256`
 - c): `cislo = 0x4FE`
 - d): `cislo = 251`
`cislo = uint8(cislo)`
 - e): `cislo = 256`
`cislo = uint8(cislo)`
 - f): `cislo = 0x4FE`
`cislo = uint8(cislo)`
- 9) Předpokládejte následující fragmenty Python programů využívajících modul numpy. Pokud bychom po každém z těchto fragmentů provedli v programu příkaz `print(cislo)`, tak co by v uvedených příkladech vypsal?
- a): `cislo = 255`
`cislo = int16(cislo)`
 - b): `cislo = 256`
`cislo = int16(cislo)`
 - c): `cislo = 255`
`cislo = int8(cislo)`
 - d): `cislo = 256`
`cislo = int8(cislo)`
 - e): `cislo = -2`
`cislo = int8(cislo)`
 - f): `cislo = -255`
`cislo = int8(cislo)`
 - g): `cislo = -3`
`cislo = int8(cislo)`
`cislo = int16(cislo)`

10) Předpokládejte následující fragmenty Python programů využívajících modul numpy. Pokud bychom po každém z těchto fragmentů provedli v programu příkaz `print(cislo, format(cislo, "02X"))`, tak co by v uvedených příkladech vypsal?

a): `cislo = 127`
`cislo = ~cislo`

b): `cislo = 127`
`cislo = ~cislo`
`cislo = uint8(cislo)`

c): `cislo = 127`
`cislo = ~cislo`
`cislo = uint16(uint8(cislo))`

d): `cislo = 127`
`cislo = ~cislo`
`cislo = uint16(cislo)`

Sada programovacích úloh – společné poznámky: Vezměte si projekt `PySerialMouseComm-SimulatedPS2StandardMouse-Shifts-ASSIGNMENT`, který odpovídá vzorovému řešení 12. úlohy ze sady self-

assessment úloh ke 3. přednášce, tj. `PySerialMouseComm-SimulatedPS2StandardMouse-COMplete-REFERENCE-SOLUTION`. Program tedy opět čte data paketů myši ve formátu „**PS/2 standard mode**“, a provádí kompletní analýzu paketu a zobrazení informace o stisknutých tlačítkách myši, a hodnoty změny v osách X a Y, a ty vypisuje jako **bezznaménková čísla** – oproti vzorovému řešení ze 3. sady úloh je program pouze rozšířen o testovací data jednoho dalšího paketu.

11) Opravte na vyznačeném místě (přiřazení do proměnných `xSA4`, a `ySA4`) kód tak, aby získal kompletní 9-bitové hodnoty X, a Y pouze **s využitím bitových operací a bitových posunů** (tj. celý výpočet hodnoty X i Y by měl probíhat **bez** potřeby jakéhokoliv větvení programu **pomocí ifů**). Hodnoty X a Y má program stále vypisovat jako **bezznaménková čísla**, tj. hodnota v proměnné `xSA4`, resp. `ySA4` má být v každé situaci stejná jako hodnota v proměnné `x`, resp. `y`. Vaše řešení nemusíte nutně zapsat jako jeden řádek, ale můžete si kód rozdělit na více řádků programu. Specifikaci komunikačního protokolu „**PS/2 standard mode**“ najdete opět v příloženém dokumentu „02-protocol (serial mice).txt“.

[Vzorové řešení viz `PySerialMouseComm-SimulatedPS2StandardMouse-Shifts-COMplete-REFERENCE-SOLUTION`.]

12) Doplněte vaše řešení úlohy 11, viz výše, o to, aby program hodnoty X a Y vypisoval správně jako **znaménková čísla**. Pozor, nechceme pouze správně vypsat, či nevypsat znak „mínus“ před hodnotu záporných čísel, ale chce mít proměnnou, ve které bude libovolná platná hodnota X, resp. Y (kterou může nám myš používající protokol „**PS/2 standard mode**“ poslat) opravdu správně jako znaménkové číslo reprezentována, a Python s ní tak bude vždy správně pracovat. K řešení můžete využít modul numpy. Pozor, že řešení nemusí být tak přímočaré, jak vypadá. V řešení této úlohy můžete použít `if`y, pokud to budete potřebovat.

[Vzorové řešení viz `PySerialMouseComm-SimulatedPS2StandardMouse-Signed-COMplete-REFERENCE-SOLUTION`.]

Poznámka: Cílem těchto úloh je, abyste si doma v rámci opakování látky z přednášky mohli sami ověřit, jak jste látku porozuměli. Úlohy jsou koncipovány víceméně přímočaře, a po projití poznámek a případně video záznamů z přednášek by jejich řešení mělo být zřejmé. Pro řešení úloh tedy není potřeba studium látky nad rámec probraný na přednáškách (nicméně je třeba mít i znalosti a pochopení látky z paralelně probíhajících přednášek a cvičení z předmětu Programování I). Pokud i po detailním a opakovaném projití látky z přednášek máte s řešením těchto úloh problém, tak se na nejasnosti co nejdříve ptejte na on-line konzultaci k přednášce, případně zvažte se mnou domluvit na konzultaci (zvlášť pokud tento stav u vás přetrvává i po dalších přednáškách).

Upozornění: Úlohy jsou vybrány a postaveny tak, abyste si po přednášce a před přednáškou následující mohli rychle připomenout hlavní části probrané látky a ověřit si její pochopení. Nicméně úlohy nejsou vyčerpávajícím přehledem látky z přednášek a tady rozhodně nepokrývají kompletní látku přednášek, která bude vyžadována u zkoušky. Pokud tedy u každé úlohy víte, jak by se měla řešit, tak to ještě neznamená, že jste dostatečně připraveni na zkoušku – nicméně jste jistě na velmi dobré cestě. Každopádně nezapomeňte, že na zkoušce se vyžaduje pochopení a porozumění právě všem konceptům ze všech přednášek, a navíc jsou zkouškové příklady postaveny komplexněji tak, aby ověřily také vaši schopnost přemýšlet nad látkou napříč jednotlivými přednáškami.

Příloha – obsah souboru 02-protocol (serial mice).txt z přednášky:

Serial mouse reset

- 1: Set UART to 'break line' state (set bit 6 in the LCR).
- 2: Clear the RTS and DTR (bits 0-1) in the MCR, wait a while.
- 3: Set the RTS and DTR bits again.

Serial mouse detection (identification bytes before optional PnP data)

In Mouse Systems mode, mouse sends nothing.
In Microsoft mode, mouse sends 'M' after dropping and raising RTS.
In Logitech mode, mouse sends 'M3' after dropping and raising RTS.
In wheel mode, mouse sends 'MZ@', 0, 0, 0 after dropping and raising RTS.

PS/2 pointing device ID (reported after 0F2h command)

In standard mode, the device reports 0.
In wheel mode, the device reports 3. This mode is enabled by sending a
Select Report Rate 200, a Rate 100 and finally a Rate 80 command sequence.
In extended mode, the device reports 4. This mode is enabled by sending a
Select Report Rate 200, a Rate 200 and finally a Rate 80 command sequence.

=====
Serial Mouse Systems mode: 1200 bps, 8 data bits, 1 stop bit, no parity

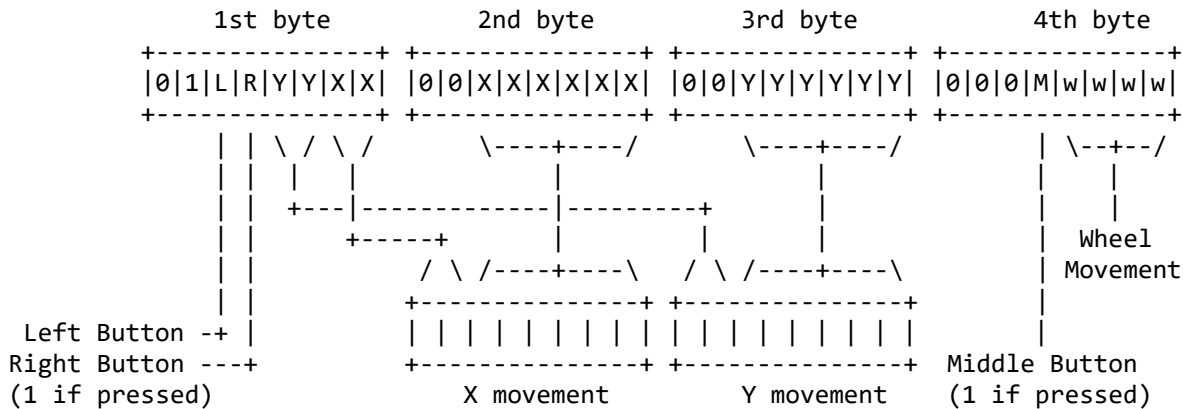
	1st byte	2nd byte	3rd byte
	+-----+	+-----+	+-----+
	1 0 0 0 0 L M R	X X X X X X X X	Y Y Y Y Y Y Y Y
	+-----+	+-----+	+-----+
		Xa movement	Ya movement
		4th byte	5th byte
	+-----+	+-----+	+-----+
Left Button	-----+	X X X X X X X X	Y Y Y Y Y Y Y Y
Middle Button	-----+		
Right Button	-----+		
(0 if pressed)		Xb movement	Yb movement

Xa/Ya - movement of the mouse since last packet.
Xb/Yb - movement of the mouse since Xa/Ya.
Movement values are 8-bit signed twos complement integers.
Positive movement value indicates motion to the right/upward.

=====

=====

Serial Microsoft wheel mode: 1200 bps, 7 data bits, 1 stop bit, no parity



First three bytes are equal to Mouse mode packet.

Movement values are 8-bit signed twos complement integers.

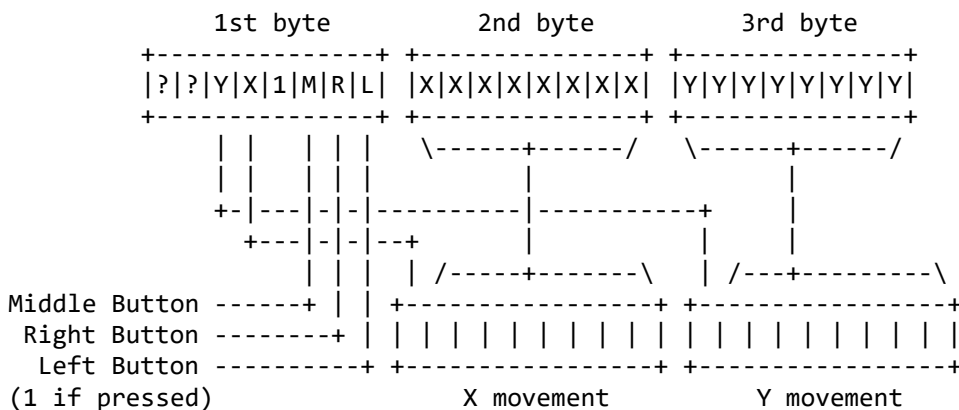
Positive movement value indicates motion to the right/downward.

Wheel movement is a 4-bit signed twos complement integer.

Positive wheel movement value indicates rotation downward.

=====

PS/2 standard mode protocol:



Two most significant bits in first byte indicate overflow (more than 9 bits of movement) in each direction. Usually ignored.

Movement values are 9-bit signed twos complement integers.

Positive movement value indicates motion to the right/upward.

=====

	1st byte	2nd byte	3rd byte	4th byte
	+-----+ +-----+ +-----+ +-----+			
	? ? Y X 1 M R L			
	+-----+ +-----+ +-----+ +-----+			
	\-----+-----/ \-----+-----/ \-----+-----/			
	+-----+ +-----+ +-----+ +-----+			
	+--- --- --- -----+-----+ +-----+ +-----+			
	+--- --- --- -----+-----+ +-----+ +-----+			
	/-----+-----\ /-----+-----\ /-----+-----\			
Middle Button	-----+ +-----+ +-----+ +-----+			
Right Button	-----+			
Left Button	-----+ +-----+ +-----+ +-----+			
(1 if pressed)	X movement	Y movement	Wheel Movement	

=====

	1st byte	2nd byte	3rd byte	4th byte
+	-----+	-----+	-----+	-----+
	? ? Y X 1 M R L	X X X X X X X X	Y Y Y Y Y Y Y Y	0 0 B F W W W W
+	-----+	-----+	-----+	-----+
		\-----+-----/	\-----+-----/	\--+--/
	+-----+-----+-----+	+-----+-----+	+-----+	+-----+
	+---+---+---+			
		/-----+-----\	/-----+-----\	
Middle Button	-----+	+-----+-----+	+-----+	+-----+
Right Button	-----+			
Left Button	-----+	+-----+-----+	+-----+	+-----+
(1 if pressed)		X movement	Y movement	(1 if pressed)

First three bytes are equal to PS/2 standard mode packet.
Two most significant bits in first byte indicate overflow (more than 9 bits of movement) in each direction. Usually ignored.
Movement values are 9-bit signed twos complement integers.
Positive movement value indicates motion to the right/upward.
Wheel movement is a 4-bit signed twos complement integer.
Positive wheel movement value indicates rotation downward.