

Odpovědi pište na zvláštní odpovědní list s vaším jménem a fotografií. Pokud budete odevzdávat více než jeden list s řešením, tak se na 2. a další listy nezapomeňte podepsat a do jejich záhlaví napsat i/N (kde i je číslo listu, N je celkový počet odevzdaných listů).

### Otázka č. 1

Předpokládejte, že máme čistě naformátovaný souborový systém používající tabulku FAT – kde, jeden záznam ve FAT má 8 bitů, hodnota 0 reprezentuje volný sektor, konec souboru reprezentuje maximální hodnota.

Souborový systém je na disku se 128 B sektory, velikost jednoho clusteru je 1 sektor. První sektor/cluster použitelný pro data souborů (tj. 1. datový sektor) je označený číslem 1, a odpovídá mu první záznam ve FAT. Souborový systém podporuje pouze jeden adresář (kořenový), a pro jeho obsah jsou napevno vyhrazeny 3 sektory před 1. datovým sektorem. Velikost jedné adresářové položky je 32 B. Každá adresářová položka obsahuje mimo jiné 11 B pro jméno souboru, 1 B pro číslo prvního sektoru, a 2 B pro velikost souboru v bytech.

Nakreslete konečný obsah prvních 16 záznamů FAT tabulky po provedení následujících operací (předpokládejte, že pokud je potřeba další volný sektor pro data souboru, tak se v souborovém systému vybere první volný sektor s nejnižším číslem; operace „zápis N bytů do X“ se chápe jako připsání N bytů za poslední byte souboru X):

- 1) Vytvoření prázdného souboru A.TXT
- 2) Vytvoření prázdného souboru B.TXT
- 3) Zápis 200 bytů do B.TXT
- 4) Zápis 50 bytů do B.TXT
- 5) Zápis 1 kB do A.TXT
- 6) Zápis 150 bytů do B.TXT
- 7) Vytvoření prázdného souboru C.TXT
- 8) Zápis 1 bytu do C.TXT

### Otázka č. 2

Předpokládejte, že máme 3 jednočipové počítače A, B, a C propojené jednou sběrnici I<sup>2</sup>C. Všechna připojená zařízení používají rychlost 100 kb/s, a 7 bitové adresování. Zařízení mají nastaveny následující adresy: A = 09h, B = 07h, C = 7Ah. Předpokládejte, že na sběrnici nikdo nevysílá, a zařízení C se rozhodne poslat zařízení A následující 2 byty: FF 55. Nakreslete a popište průběh napětí (a jejich logických hodnot) na vodičích Serial Data Line (SDA) and Serial Clock (SCL) od začátku až do konce výše zmíněného přenosu (předpokládejte, že během přenosu nedojde k žádným chybám).

### Otázka č. 3

Napište program v jazyce Pascal (případně v jazyce C), který na standardní výstup (tj. pomocí procedury WriteLn) vypíše text „Little Endian“ nebo „Big Endian“ bez uvozovek podle toho, na jaké platformě bude spuštěn (resp. pro kterou bude přeložen). Připomenutí: prefixový unární operátor @ slouží v Pascalu pro získání adresy libovolné proměnné.

### Otázka č. 4

Spočítejte hodnotu následujícího výrazu zapsaného v Pascalu (předpokládejte, že celý výpočet i všechny uvedené hodnoty jsou v 64-bitových celých číslech bez znaménka):

$$(10 \text{ OR } \$2000) \text{ XOR } (\$3 \text{ SHL } 1)$$

Výsledek zapište jako jedno celé číslo v desítkové soustavě.

### Otázka č. 5

Předpokládejte, že implementujete funkci OS, která pošle N bytů dat po lokální síti. Funkce OS dostane od aplikace v jednom parametru ukazatel na první byte, který má po síti poslat, a ve druhém parametru číslo N. Pro komunikaci se síťovou kartou (pro zápis do jejích registrů) se používá mechanismus *port-mapped IO*, a síťová karta používá mechanismus DMA pro přenos dat z/do hlavní paměti RAM. Před začátkem zápisu tedy musí OS do registrů síťové karty zapsat zdrojovou adresu v paměti RAM, která ukazuje na data připravená pro odeslání. Kvůli optimalizaci výkonu bychom ale chtěli podporovat jen situaci, kdy síťová karta vždy získá přímo adresu původních dat ve zdrojové aplikaci (tj. před provedením operace odeslání nikdy nechceme data kopírovat na jiné místo v paměti). Za předpokladu, že se v OS používá mechanismus stránkování, vysvětlete následující:

- a) Vysvětlete, co vše musí OS v takové situaci udělat, aby síťové kartě předal správnou adresu. Na příkladu uveďte jakou.
- b) Bude tento princip fungovat pro všechny možné adresy a hodnoty N, které může aplikace operačnímu systému předat? Vysvětlete proč ano, resp. proč ne.

### Otázka č. 6

Následující obrázek obsahuje část screenshotu hex editoru, který zobrazuje obsah 52 bytů dlouhého binárního souboru:

	0001	0203	0405	0607	0809	0A0B	0C0D	0E0F
00	0000	0000	0000	F07F	0000	0000	0000	F0FF
10	7200	E100	6400	0000	67C1	5801	ED00	7000
20	2000	6A00	6500	2000	6B00	6F00	7000	6500
30	6300	2E00						

Víme, že všechna data jsou v souboru uložena jako little endian, a že od 22. bytu (počítáno od 0) je v souboru uloženo 32-bitové reálné číslo s pohyblivou desetinnou čárkou. Mantisa je normalizována se skrytou 1 a zabírá spodních 23 bitů, pak následuje 8-bitový exponent uložený ve formátu s posunem (bias) +127 a 1 znaménkový bit. Zapište hodnotu tohoto reálného čísla v desítkové soustavě.

**Otázka č. 7**

Předpokládejte, že v OS s podporou pro vícevláknové zpracování dojde k náhlému ukončení nějakého vlákna (např. po dereferenci neplatného ukazatele) v situaci, kdy toto vlákno drží několik zamčených zámků. Implementace zámků je poskytována operačním systémem. Jak se v takové situaci má OS zachovat? Popište všechny typické možnosti řešení daného problému a vysvětlete jejich výhody a nevýhody.

**Otázka č. 8**

Předpokládejme následující část programu v jazyce Pascal (jednotlivé řádky programu v Pascalu jsou očíslované a označené *kurzívou*; pod každým řádkem v Pascalu jsou vypsané instrukce procesorové řady x86, na prvním řádku jsou vždy zapsané byty strojového kódu dané instrukce, na druhém řádku je pak v odsazení uveden zápis dané instrukce v Intel assembleru; proměnné a, b, c jsou typu Longint):

```
ř15: a := a + b;
      A1 20 C0 40 00
          mov    eax, [0040C020h]
      8B 15 30 C0 40 00
          mov    edx, [0040C030h]
      01 D0
          add    eax, edx
      A3 20 C0 40 00
          mov    [0040C020h], eax
ř16: b := 0;
      C7 05 30 C0 40 00 00 00 00 00
          mov    [0040C030h], 0
ř17: c := a + 8;
      A1 20 C0 40 00
          mov    eax, [0040C020h]
      83 C0 08
          add    eax, 8
      A3 40 C0 40 00
          mov    [0040C040h], eax
```

Nyní předpokládejme, že chceme tento program ladit a na řádek číslo 17 umístit breakpoint. V tomto kontextu odpovězte na následující otázky:

- Je třeba, aby debugger rozuměl zdrojovým kódům jazyka Pascal? A pokud ne, jak debugger pozná, že má vykonávání programu zastavit zrovna před provedením instrukce `mov eax, [0040C020h]`?
- Jak debugger způsobí, že se program „zastaví“ před provedením instrukce `mov eax, [0040C020h]`, když přeci procesor stále musí nějaký kód vykonávat?

**Otázka č. 9**

Následující program zapsaný v jazyce Pascal můžete pomocí překladače Free Pascal přeložit a spustit jak na OS Linux na platformě x86, tak i na OS Windows 8 na platformě x86. Popište a vysvětlete, z jakého důvodu je to možné a jaké všechny kroky je třeba provést, abyste z původního zdrojového souboru získali spustitelný soubor. Do výkladu zahrňte zdůvodnění, zda a proč pro daný scénář bude potřeba více různých spustitelných souborů daného programu, nebo zda a proč bude dostačovat pouze jeden.

```
program Mocnina;
var
  m, n : integer;

begin
  ReadLn(n);
  m := 1;
  while n >= 1 do begin
    m := m * 2;
    Dec(n);
  end;
  WriteLn('2 na N je ', m);
end.
```

**Otázka č. 10**

Naimplementujte v Pascalu funkci `DelkaTextu` s následujícím prototypem:

```
type
  PUtf32 = ^longword;

function DelkaTextu(text : PUtf32)
  : longword;
```

která jako parametr `text` bere ukazatel na null-terminated řetězec v kódování UTF-32, a vrátí počet kompletních znaků (grafémů), ze kterých se tento řetězec skládá (tedy např. pro libovolný vstupní řetězec reprezentující text Říp má funkce `DelkaTextu` vrátit hodnotu 3). Předpokládejte, že velikost typu `longword` jsou 4 byty). Pokud byste od RTL nutně potřebovali nějakou pomocnou funkci nebo proceduru, tak si ji zadeklarujte, a popište chování, které od ní očekáváte.