

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

Následující procedura DoMap naprogramovaná v jazyce Pascal zpracuje všechny položky pole předaného v parametru data tak, že na každou zavolá proceduru jejíž adresa jí byla předána v parametru mapProc:

**type**

```
TValueArray = array[0..N-1] of Pointer;
PValueArray = ^TValueArray;
PMapProc = procedure (value : Pointer);
```

**procedure** DoMap(  
 data : PValueArray; mapProc : PMapProc  
 );

**var**  
 i : Integer;

**begin**

```
for i := 0 to Length(data^) - 1 do begin
  mapProc(data^[i]);
end;
```

**end;**

**end;**

Předpokládejte, že proceduru DoMap budeme používat pouze v programech implementovaných pro nějaký konkrétní OS s podporou vícevláknového zpracování, a že ji budeme často volat na víceprocesorovém (resp. vícejádrovém) systému. Napište novou lepší implementaci procedury DoMap (v Pascalu, případně v jazyce C) tak, aby na nevytíženém systému plně využívala výkonu všech dostupných procesorových jader. Předpokládejte, že na pořadí volání procedury mapProc na jednotlivých položkách pole data nezávisí, a že celá procedura DoMap smí skončit, až když byla volání procedury mapProc dokončena na všech položkách pole data.

Všechny potřebné API funkce/procedury OS pro podporu vícevláknového zpracování si vhodně navrhnete sami. Pro každou takovou funkci/proceduru OS jen stručně popište, co dělá, ale neimplementujte ji.

### Otázka č. 2

Předpokládejte X-bitový procesor s A-bitovým paměťovým adresovým prostorem (kde, šířka vnější adresové sběrnice je A bitů, a šířka vnější datové sběrnice je D bitů). Např. předpokládejte procesor, kde A=36, D=16, X=32. V tomto kontextu odpovězte na následující:

- Nejprve obecně a pak na uvedeném příkladu vysvětlete, co je slovo (word) a jaká je jeho velikost?
- Velmi často se můžete setkat s tím, že jako velikost jednoho slova se chápe konkrétní konstanta, která nemusí odpovídat obecné definici slova podle bodu a). Napište, jaká tato konstanta je, a vysvětlete, proč se používá.

### Otázka č. 3

Popište, co vše zahrnuje a určuje volací konvence procedur a funkcí. Uveďte příklad jedné běžně používané volací konvence a detailně ji popište.

### Otázka č. 4

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. 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:

- Vysvětlete, co vše musí OS v takové situaci udělat, aby síťové kartě předal správnou adresu. Na příkladu uveďte jakou.
- 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. 5

Srovnejte termíny *cluster* a *cloud* a popište jejich výhody a nevýhody.

### Otázka č. 6

Následující obrázek obsahuje část screenshotu hex editoru, který zobrazuje obsah 68 bytů dlouhého binárního souboru:

	0001	0203	0405	0607	0809	0A0B	0C0D	0E0F
00	0000	0000	0000	F07F	0000	0000	0000	F0FF
10	5000	5901	ED00	6C00	6900	6101	2000	7E01
20	6C00	7500	6501	6F00	7500	0D01	6B00	FD00
30	2000	6B00	6F01	4801	2E00	E000	80C7	7200
40	E100	6400						

Víme, že všechna data jsou v souboru uložena jako little endian, a že od 58. 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**

Vysvětlete, co je instrukce podmíněného skoku (conditional jump/branch), a v jakých situacích se používá. Dále uveďte nějaký příklad, jak taková instrukce může na typickém procesoru vypadat a jaké bude mít argumenty – pro takový příklad instrukce popište, jak se bude procesor chovat ve všech možných situacích, které mohou nastat při jejím zpracování.

---

**Otázka č. 8**

Vysvětlete a popište všechny akce, které se dějí v procesoru při vzniku hardwarového přerušení až do jeho kompletního obslužení. Do výkladu zahrňte, zda se průběh zpracování nějak liší při vzniku NMI.

---

**Otázka č. 9**

Předpokládejte počítač s 32-bitovým paměťovým adresovým prostorem. Architektura počítače zahrnuje zvláštní adresový prostor pro komunikaci mezi procesorem a připojenými zařízeními a v systému se nepoužívá žádné memory-mapped I/O. V systému je nainstalována 512 kB velká paměť ROM, která je souvisle namapována na nejvyšší možné adresy v paměťovém adresovém prostoru počítače. Dále je v systému nainstalováno N bytů paměti RAM, která je souvisle namapována od adresy 0 v paměťovém adresovém prostoru počítače. Máte jistotu, že součet velikostí nainstalované paměti RAM a paměti ROM nepřesahuje velikost paměťového adresového prostoru.

Předpokládejte, že v Pascalu (případně v jazyce C) implementujete část firmware počítače, který bude uložený ve zmíněné paměti ROM. Napište implementaci funkce s následujícím prototypem (viz níže), která vrátí velikost instalované paměti RAM (tj. vrátí číslo N). Předpokládejte, že překladač Pascalu generuje pro vaši funkci takový strojový kód, že pro všechny lokální proměnné používá pouze registry procesoru a nepotřebuje použít žádnou část paměti RAM. Předpokládejte, že v průběhu celé vaší funkce jsou zakázána všechna přerušení a NMI nemůže vzniknout. Předpokládejte, že není zapnutá segmentace, ani stránkování. Předpokládejte, že typ Longword slouží pro ukládání bezznaménkových celých čísel a jeho velikost je 32-bitů.

**function** GetAvailableMemory : Longword;

---

**Otázka č. 10**

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.