

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. Do zápatí všech listů vždy napište i/N (kde i je číslo listu, N je celkový počet odevzdaných listů).

Společná část pro otázky označené X

Předpokládejte, že máme sběrnici, která je variantou původní 8-bitové sběrnice ISA, tedy paralelní sběrnice nepodporující burst přenosy, ani DMA bus mastering. Navíc má ale naše varianta následující odlišnosti: podpora pouze pro 4-bitový paměťový adresový prostor, a pro 4-bitový I/O adresový prostor, základní taktovací frekvence je stejná jako u nejběžnější základní varianty sběrnice PCI.

Otázka č. 1 (X)

Nakreslete časový diagram průběhu hodnot na všech důležitých vodičích v průběhu jedné celé transakce zápisu hodnoty \$0E na adresu \$4 v paměťovém adresovém prostoru. Nakreslete průběh pro každý takový vodič zvlášť, vodiče neseskupujte. K vašemu obrázku napište krátkou legendu, tj. pro každý použitý vodič vysvětlíte jeho význam.

Otázka č. 2 (X)

Předpokládejte, že budeme pomocí výše uvedené sběrnice zapisovat 1 MB dat do 16-bitového datového registru řadiče síťové karty – pro přístup k registru se používá MM I/O, registr leží na adresách \$2, a \$3. Předpokládejte, že data budeme zapisovat maximální rychlostí sběrnice, a v průběhu celé operace se nebude na sběrnici vyskytovat žádný další provoz. Pro uvedený scénář určete přibližně průběrnou přenosovou rychlost sběrnice v kB/s. Zapište a vysvětlíte i postup výpočtu!

Otázka č. 3

Předpokládejte, že máme na počítači s procesorem Intel 8088 kompatibilním nainstalován operační systém s kooperativním přepínáním vláken a se všemi potřebnými ovladači zabudovanými v jádře. Na systémové sběrnici počítače je připojen jednoduchý USB 1.1 řadič s následujícím HCI: Řadič vždy odesílá/přijímá sekvenci paketů, které jsou uspořádány do jednosměrně vázaného seznamu jejich descriptorů (každý descriptor vždy obsahuje informace o jednom paketu, tj. o adrese jeho dat, resp. adrese kam se jeho data mají uložit po příjmu, o směru přenosu paketu, apod.). Do řadiče je vždy (mimo jiné) naprogramována adresa descriptoru prvního z USB paketů k odeslání/příjmu. Vždy po odeslání/příjmu kompletní naprogramované sekvence paketů vyvolá USB řadič maskovatelné přerušování 3.

Dále předpokládejte, že v systému dojde k zacyklení aktuálně běžícího vlákna, které ale v daném cyklu již nikdy nevyvolá proceduru Yield. Pokud je k počítači připojena pouze USB klávesnice, je možné OS naprogramovat tak, aby bylo možné donutit takové zacyklené vlákno se přepnout na jiné po stisku klávesy Break? Pokud ne, tak vysvětlíte proč. Pokud ano, popište a vysvětlíte, co se bude v systému dít od samotného stisku klávesy Break, až do vyvolání procedury Yield, která provede přeplánování na jiné vlákno (do popisu zahrňte i to, jak se OS dozví o stisku klávesy).

Otázka č. 4

Předpokládejte, že máme počítač s variantou 32-bitového procesoru Intel 80386 běžícího v 32-bitovém režimu s vypnutou podporou pro stránkování (virtuální i fyzický adresový prostor procesoru má šířku 32-bitů, virtuální adresa se přímo rovná adrese fyzické). Procesor má obecnou registrovou architekturu, a mimo jiné 5 obecných 32-bitových registrů EAX, EBX, ECX, EDX, EBP, dále má 32-bitový registr ESP (stack pointer), a 32-bitový registr EIP (program counter). Dále víme, že od adresy 00CA19EEh je v paměti uložen kód funkce DoSomeMagic, která má dva 32-bitové unsigned parametry, a vrátí 32-bitovou unsigned hodnotu. Funkce používá variantu Cčkově volací konvence (parametry se předávají na zásobníku zprava doleva, parametry odstraňuje volající, návratová hodnota se ukládá do registru EAX).

Dále máme k dispozici disassembler generující kód ve variantě Intel syntaxe, tedy: cílový operand instrukce je vždy nejvíce vlevo; hodnota v hranatých závorkách znamená variantu instrukce s operandem typu adresa, tj. např. [x] znamená hodnotu operandu na adrese x; běžné instrukce procesoru 80386 mají i variantu, kde se adresa operandu spočítá jednoduchým výpočtem, a tedy např. zápis [x+y] znamená operand ležící na adrese, která vznikne součtem hodnot x a y.

Po spuštění disassembleru na adresu 00CA19EEh jsme zjistili, že kód funkce DoSomeMagic je následující:

```
00CA19EE sub    esp, 4
00CA19F1 mov    eax, [esp+8]
00CA19F5 add    eax, 0000000Ah
00CA19F8 mov    ebx, [esp+0Ch]
00CA19FC cmp    eax, ebx
00CA19FE jnz   0CA1A09h
00CA1A00 mov    [esp], 00000005h
00CA1A07 jmp   0CA1A10h
00CA1A09 mov    [esp], 00000008h
00CA1A10 mov    eax, [esp]
00CA1A13 add    esp, 4
00CA1A16 ret
```

Zapište v Pascalu její kód i její kompletní deklaraci bez použití inline assembleru. Jména proměnných a parametrů, která nejsou z disassemblovaného kódu zřejmá, si vhodně zvolte. Předpokládejte, že typ Longword je 32-bitové celé číslo bez znaménka.

Otázka č. 5

Popište a vysvětlíte, jakým způsobem se typicky překládají a spouštějí programy napsané v Javě nebo v jazyce C#. Do svého vysvětlení zahrňte, co to je, a jaký význam a výhody má v tomto kontextu tzv. *intermediate language*.

Otázka č. 6

Procesor Intel 8086 je 16-bitový procesor s 16-bitovou datovou sběrnicí a 20-bitovou adresovou sběrnicí (podpora pro 20-bit paměťový a 16-bit I/O adresový prostor) a s typickou taktovací frekvencí 4,77 MHz. Procesor má celkem 20 adresově/datových pinů, kde je ve dvou taktech procesoru časově multiplexován přenos adresy a dat. Procesor také podporuje i tzv. *nezarovnané přístupy* k datům v paměti. Vysvětlete, co v tomto kontextu tento termín znamená, a popište, zda má podpora nezarovnaných přístupů nějaké výhody, či nevýhody.

Otázka č. 7

V operačním systému Linux se pro spustitelné soubory používá formát ELF (Executable and Linkable Format). Spustitelné soubory v tomto formátu obsahují i tzv. *relokační tabulku*. Vysvětlete, co tento termín znamená, co je obsahem typické relokační tabulky, kdo její obsah generuje, a kdo a proč její obsah čte. Do vysvětlení zahrňte i to, zda je vůbec tento koncept potřebný i v operačním systému Linux, který používá mechanismus stránkování pro oddělení adresových prostorů běžících procesů.

Otázka č. 8

Předpokládejte následující program, který se bez chyb přeloží překladačem FreePascalu:

```

program Prg;
type
  TPole = array[1..10] of Longword;
  PLongword = ^Longword;

procedure Aktualizuj(x : PLongword);
begin
  while x^ <> 0 do begin
    x^ := x^ + 1;
    x := x + 2;
  end;
end;

var
  p : TPole;
  i : Integer;

begin
  for i := 1 to 10 do
    p[i] := 4;
  p[5] := 0;
  Aktualizuj(@(p[1]));
  for i := 1 to 10 do
    Write(p[i], ', ');
end.

```

Co tento program po spuštění vypíše na standardní výstup? Předpokládejte, že typ Longword je 32-bitové celé číslo bez znaménka.

Otázka č. 9

Předpokládejte, že v programu potřebujeme pracovat s velkým množstvím 512 položkových sekvencí boolových hodnot – chceme si proto naimplementovat maximálně paměťově úspornou datovou strukturu TBitovePole (implementující tzv. bitové pole), kde se na uložení každé boolovské hodnoty (true nebo false) použije právě jednoho bitu. Níže uvedená procedura Nastav slouží ke změně konkrétního bitu v takovém bitovém poli: její parametr index je číslo měněného bitu od 0 do 511; parametr nova obsahuje novou hodnotu, která má být na dané pozici nastavena (ostatních 511 bitů, na které se parametr index neodkazuje, musí samozřejmě zůstat beze změny).

```

const
  Max = 16;
type
  TBitovePole = array[1..Max] of Longword;

procedure Nastav(
  var pole : TBitovePole;
  index : Longword; nova : Boolean
);

```

Naprogramujte proceduru Nastav tak, aby její chování odpovídalo výše uvedenému popisu a deklaracím. Předpokládejte, že typ Longword je 32-bitové celé číslo bez znaménka.

Otázka č. 10

Předpokládejte, že k 32-bitovému procesoru Intel IXP425 s architekturou ARM pracujícímu na frekvenci 533 MHz chceme připojit řadič grafického LCD displeje s Full HD rozlišením 1920*1080 pixelů. Řadič má v sobě integrováno 6 MB paměti typu SRAM, která slouží jako grafická paměť (video RAM) (pro každý z 1920*1080 pixelů obsahuje 3 byty informace o jeho barvě reprezentované jako RGB [pro informace o intenzitě každé barevné složky je použito jedno 1 bytové beznaménkové celé číslo]). Pro tento řadič navrhujeme jeho HCI, které má kódu běžícímu na uvedeném procesoru umožnit změnu části zobrazované informace pomocí přímého zápisu do grafické paměti. Pro přístup ke grafické paměti se má používat mechanismus MM I/O.

Řadič LCD displeje a uvedený procesor budeme propojovat pomocí paralelní multidrop sběrnic, která podporuje 24-bitové adresy v paměťovém adresovém prostoru. Na sběrnicí budou ale připojena i další zařízení s velkou spotřebou adres pro jejich HCI, proto potřebujeme obsazení adresového prostoru řadičem LCD minimalizovat. Je možné HCI řadiče LCD navrhnout tak, aby zabíralo maximálně 1/8 adresového prostoru, ale zároveň bylo možné změnit zobrazovaná data v libovolné části displeje (tedy bylo možné provést z kódu běžícího na procesoru zápis na libovolnou adresu ve video RAM)? Pokud ne, vysvětlete proč. Pokud ano, tak takové HCI navrhnete, a popište, jak bude komunikace s řadičem (zápis na nějaké místo ve video RAM) probíhat.