

Elektronická pomůcka pro předmět Mikropočítače

Lenka Novosádová

2005



Univerzita Tomáše Bati ve Zlíně
Fakulta technologická

Vloženo oficiální zadání

ABSTRAKT

Abstrakt česky

Cílem práce bylo vytvořit dílčí prostředky pro elektronickou podporu výuky předmětu Mikropočítače. Je zpracován úvod do problematiky mikropočítačů se zaměřením na konkrétní mikropočítač firmy Motorola HC11. Jsou popsány součásti mikropočítače - centrální procesorová jednotka, paměť dat a programů, komunikační kanály a vstupně výstupní porty.

Stěžejní část práce je obsažena v druhé části, ve které byly porovnány dvě aplikace pro vytváření testů pro kontrolní činnost zvládnutí přednášené látky posluchači. Tyto aplikace jsou porovnány a je doporučena k použití jedna z nich - prostředí Moodle. Tento způsob testování studentů může sloužit jednak pro přípravu studentů na zkoušku a jednak umožňuje získat přehledové znalosti studentů pro celkové závěrečné hodnocení. V rámci práce jsou dále zpracovány vybrané kapitoly formou prezentace v prostředí Power Pointu.

Abstrakt ve světovém jazyce

The aim of the work was to create the education electronic support sub-framework for the subject Microcomputers. There is processed the introduction into the microcomputers aimed at the particular microcomputer HC11 produced by Motorola company. There are the parts of microcomputer described – the central processing unit (CPU), the memory system, the bus system and I/O ports.

There is the pivot part in the second part of the work. There are two creating tests applications confronted. Finally for the disposal is recommended the database Moodle. This way of students testing should be used to the student's revise and to the getting comprehensive knowledges for the final exams. There are selected chapters processed as the presentation while using the Microsoft PowerPoint.

Děkuji prof. Ing. Vladimíru Vaškovi, CSc., vedoucímu bakalářské práce, za odborné vedení, cenné rady a připomínky, které mi pomohly k jejímu dokončení.

OBSAH

OBSAH.....	5
ÚVOD.....	6
1 Mikropočítače.....	7
1.1 Vývoj mikropočítačů.....	7
1.2 Základní pojmy.....	7
1.2.1 Mikroprocesor.....	7
1.2.2 Jednočipový mikropočítač.....	8
1.2.3 Systémová sběrnice.....	8
1.2.4 Registr.....	8
1.2.5 Paměť mikropočítače.....	9
1.3 Mikroprocesor HC11.....	9
1.3.1 Součásti HC11.....	12
1.3.2 Systém přerušení.....	13
1.3.3 Programování HC11.....	15
1.4 Operační systémy pro řízení v reálném čase.....	17
1.4.1 Stavy procesů.....	18
1.4.2 Plánování procesů.....	19
1.4.3 Průběh vykonání procesu.....	19
1.4.4 Reálný čas a mikropočítače.....	20
2 testovací programy.....	23
2.1 Testy utb.....	23
2.1.1 Aplikace TESTY.....	23
2.1.2 Úvodní stránka.....	24
2.1.3 Psát test.....	25
2.1.4 Správa testů.....	27
2.1.5 Volné testy.....	35
2.2 Testy Moodle.....	36
2.2.1 Úvod k Moodle.....	36
2.2.2 Úvod k tvorbě testů.....	37
2.2.3 Typy úloh.....	39
2.2.4 Nastavení úlohy.....	43
2.2.5 Zavedení testu do systému.....	48
2.3 Srovnání testovacích programů.....	53
2.3.1 Výhody a nevýhody aplikace Testy.....	53
2.3.2 Výhody a nevýhody vytváření testů v systému Moodle.....	53
2.3.3 Výběr aplikace.....	54
ZÁVĚR.....	55
SEZNAM POUŽITÉ LITERATURY.....	56
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	57
SEZNAM OBRÁZKŮ.....	58
SEZNAM TABULEK.....	59
SEZNAM PŘÍLOH.....	60

ÚVOD

Dříve byl mikroprocesor chápán jako programovatelný sekvenční automat, pro který byla charakteristická krátká délka datového slova (4, 8 bitů), malý počet instrukcí, malé adresové prostory a to jak pro program a data, tak pro vstupně výstupní periferie. Označoval tedy programovatelný obvod jednoduché konstrukce s nepříliš vysokým výpočetním výkonem, určený k měření základních řídicí a regulačních úloh. Dnes je již mikroprocesor chápán jako programovatelný sekvenční automat s výkonem velkého sálového počítače. Jednoduché programovatelné sekvenční automaty s integrovanými vstupně výstupními periferiemi na jednom čipu jsou označovány jako mikropočítač či jednočipový mikropočítač. Označení mikrořadič či mikrokontroler pak používáme pro nejjednodušší zástupce mikropočítačů.

Jednočipové mikropočítače zpravidla obsahují oddělenou paměť programu (zpravidla užívaný název pro ROM, PROM nebo EPROM mikrořadiče) a paměť dat (zpravidla užívaný název pro RAM mikrořadiče) a časovače, které umožňují synchronizaci s vnějším okolím - s reálným světem. Proto se někdy jednočipovým mikropočítačům říká řídicí systémy v reálném čase. Aplikační rozsah těchto řídicích systémů je prakticky neomezený od běžných elektronických přístrojů používaných v domácnosti jako je videorekordér, automatická pračka apod. až po nejnáročnější aplikace v automobilovém a leteckém průmyslu. Největšího rozšíření z hlediska aplikací doznaly mikrokontroléry dvou významných výrobců mikroprocesorů - firem Motorola a Intel.

Úkolem bakalářské práce je uvedení do problematiky a seznámení se základními pojmy spojenými s mikropočítači. Zaměříme se především na osmibitový mikropočítač HC11 firmy Motorola. Součástí bakalářské práce je také prostudování programového vybavení pro testování znalostí studentů z problematiky předmětu Mikropočítače. Test může být vytvořen ve dvou již existujících aplikacích. Porovnání těchto aplikací rozhodneme, která z nich bude pro náš test vhodnější.

1 MIKROPOČÍTAČE

1.1 Vývoj mikropočítačů

Nejdůmyslnější stroj, který člověk doposud vymyslel, je počítač. Mikropočítač je určitým druhem počítače. První integrovaný obvod, který můžeme nazvat mikroprocesor, navrhl pracovník firmy INTEL, E. Hoff, při práci na vývoji obvodů pro stolní kalkulátor. Napadlo jej přenést na programovatelnou strukturu univerzálního procesoru řešení úloh, které se původně předpokládalo řešit sekvenčním automatem. Výsledný obvod byl nazván mikroprocesor a označen 4004. Měl šířku dat 4 bity. Instrukční soubor orientovaný hlavně na práci s čísly v BCD kódu a na logické funkce, obsahoval 2 300 tranzistorů. Vývoj pokračoval dále za účasti Federico Faggina a Massatoshi Shimmy (zakladatelů firmy ZILOG) a byla vytvořena mikropočítačová sada MCS4, která byla nabídnuta k obecnému použití. Úspěch byl překvapující. Za 3 měsíce bylo prodáno na 85 000 sad a prokázalo se, že doba dozrává pro zavedení programovatelných automatů typu mikropočítač. Začátkem roku 1974 pak firma INTEL přišla s typem 8080, který se stal na delší dobu jakýmsi světovým standardem. V ČSR se vyráběl pod označením MHB8080A. Asi rok po zavedení typu 8080 uvedla na trh firma MOTOROLA mikroprocesor MC6800 a v roce 1976 firma ZILOG mikroprocesor Z80-CPU. Toto byli hlavní představitelé osmibitových mikroprocesorů. Od roku 1978 jsou středem pozornosti 16bitové mikroprocesory. Mikroprocesor 8086 firmy INTEL uvedený na trh v roce 1978 byl vyroben technologií HMOS s 29 000 tranzistory. S ročním zpožděním nabídla firma ZILOG mikroprocesor Z8000 a firma MOTOROLA mikroprocesor MC68000, který se navenek jeví jako 16bitový, ale vnitřní strukturu má z velké části 32bitovou. Koncem 70. let nevznikaly jen 16bitové mikroprocesory, ale i monolitické mikropočítače, které představují samostatnou větev vývoje od osmibitového obvodu 8048 až po 16bitové jako např. 80186. Vývoj samozřejmě pokračuje stále kupředu přes transputery a procesory typu RISC. V současné době jsou v počítačích používány 64bitové procesory. [1, 2, 10, 15]

1.2 Základní pojmy

1.2.1 Mikroprocesor

V každém počítači nalezneme jeden či více mikroprocesorů. Jedná se o část zajišťující přesuny dat v počítači a jejich zpracování. Mikroprocesor je základní částí mikropočítače. V podstatě jde o centrální procesorovou jednotku (CPU). Mikroprocesor

obsahuje aritmeticko-logickou jednotku (ALU). Jde o sčítačku doplněnou posuvnými registry a logickými obvody. Vykonává operace spojené se zpracováním dat (matematické, logické a posuvy). Počet bitů, se kterými je schopna ALU pracovat, udává, kolikabitový je celý mikroprocesor. Musí obsahovat také pracovní registry. Jednotlivé mikroprocesory se od sebe liší počtem registrů a jejich velikostí, která udává jak velké číslo jsme schopni v něm uchovat. Vstupní a výstupní registr pro hodnoty určené ALU označujeme střádač. Důležitou součástí jsou také obvody řízení. Zajišťují vykonání instrukce vytvořením posloupnosti impulsů, která ovlivní jednotlivé části procesoru tak, aby po ukončení této posloupnosti byla instrukce vykonána. [1, 12, 16]

1.2.2 Jednočipový mikropočítač

Jednočipový mikropočítač, označován také jako mikrokontrolér, mikrořadič (anglicky embedded microcontroller), který je svou architekturou přizpůsoben speciálně pro monitorování a řízení různých mechanismů a procesů. Kromě vlastního mikroprocesoru jsou na čipu integrovány ještě operační paměť (RAM), pevná paměť (ROM, PROM nebo EPROM), kde je uložen řídicí program, vstupní a výstupní obvody, případně přímo potřebná rozhraní pro řízení periférií, například seriové rozhraní, časovač, A/D nebo D/A převodníky, výkonové budiče řídicích číslicových signálů apod.. Časovače umožňují synchronizaci s vnějším okolím - s reálným světem. Proto se někdy jednočipovým mikropočítačům říká řídicí systémy v reálném čase. [1, 15]

1.2.3 Systémová sběrnice

Jde o soustavu vodičů, která slouží k transportu dat, řídicích signálů a adres mezi mikroprocesorem, pamětí a vstupně/výstupními obvody. Skládá se ze tří částí. *Datová sběrnice* slouží k přenosům dat a kódů instrukcí. *Řídicí sběrnice* synchronizuje všechny části počítače. Poslední součástí je *adresová sběrnice*, určená k přenosům adres slabik v paměti a adres vstupně/výstupních portů. [1, 2, 16]

1.2.4 Registr

Představuje paměťovou buňku, vytvořenou přímo ve struktuře mikroprocesoru. Využívá se k dočasnému uchování informací v průběhu vykonávání operací nebo je využíván programem. Různé typy mikroprocesorů se liší různým počtem registrů. Mezi základní registry patří *čítač instrukcí*. Obsahuje adresu instrukce, která se má vyjmout z hlavní paměti a posléze procesorem provést. Dále *univerzální (pracovní) registry*, které

mívají délku jednoho slova. Do těchto registrů se ukládají data, které slouží jako operandy při provádění některých instrukcí. *Indikační registry* bývají obvykle jednobitové, které indukují určitý stav (např. příznak znaménka, přenosu atd.). V registru, označovaném *ukazatel na zásobník*, je uložena adresa vrcholu zásobníku. [1, 16]

1.2.5 Paměť mikropočítače

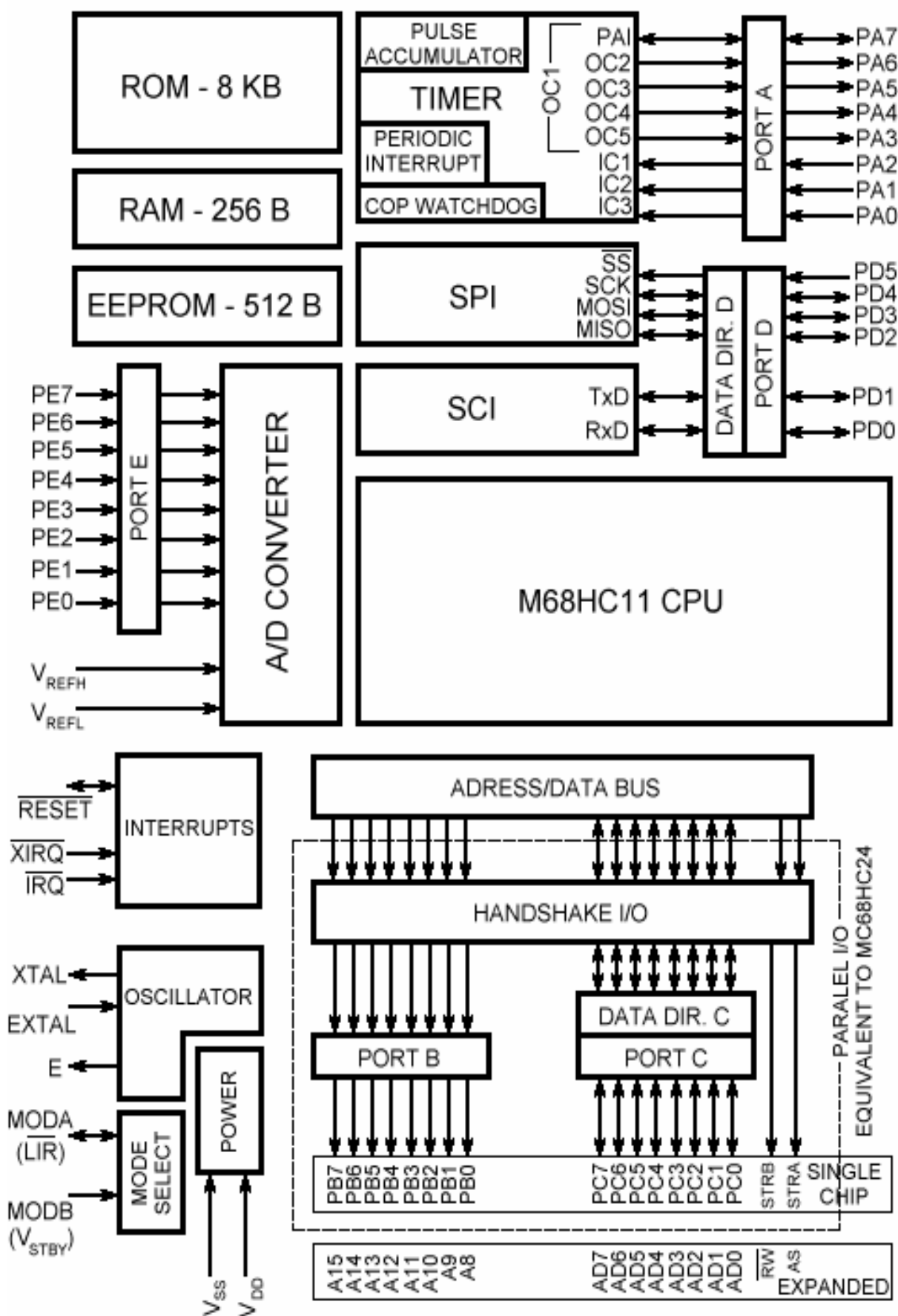
Dělíme na paměť programů (pevnou) a paměť dat. Paměť dat označujeme zkratkou z anglického termínu Random Access Memory, tedy *RAM*. Tato paměť zajišťuje dočasné uložení dat zpracovávaných mikroprocesorem. Data do paměti může mikroprocesor uložit a opět zpětně vyzvednout. Do této paměti tedy lze také zapisovat.

Pevná paměť označována *ROM* z anglického termínu Read Only Memory. Z této paměti lze pouze číst, je naprogramována již při výrobě. Obsahuje ve většině případů instrukce, které zajišťují realizaci daného algoritmu řízení pro přizpůsobení mikropočítače určité aplikaci. Paměť může také obsahovat konstanty a neměnné tabulky používané v programu. Určitými variantami paměti ROM jsou paměti PROM, EPROM a paměť FLASH. [1, 12, 13, 14, 16]

1.3 Mikroprocesor HC11

Mikropočítač MC68HC11 je snad nejrozšířenějším mikropočítačem v malých řídicích aplikacích. Počátek vývoje řady M68HC11 představuje známý osmibitový mikroprocesor M6801, vyráběný od počátku osmdesátých let technologií HMOS. Jeho vylepšená verze v provedení CMOS slouží nyní jako jádro mikroprocesorů řady M68HC11. Řadu M68HC11, dnes představující více než dvacet různých typů mikroprocesorů, nabízí široké možnosti zvolit čip, který optimálně vyhovuje právě pro uvažovanou aplikaci. Tyto typy se navzájem liší velikostí a druhem pamětí, které jsou s jádrem naintegrované na jeden čip také druhem a počtem kanálů periférií, které jsou na tomto čipu jádrem přímo ovládány. Jednočipové mikroprocesory řady M68HC11 byly původně vyvinuty pro oblast standardních aplikací v automobilovém průmyslu. Jejich hlavní přednosti jsou vícekanálové analogové vstupy, promyšlený systém časovačů a paměť typu EEPROM přímo na čipu. Jsou však velmi vítanými vlastnostmi i v jiných oblastech. Díky vysoké spolehlivosti mikroprocesorů M68HC11 v širokém rozsahu provozních teplot a příznivé ceně dosahuje počet prodaných kusů ve světě řádově stovek miliónů. Každým rokem reaguje firma Motorola na zájem zákazníků několika novými

typy, které vhodně doplňují vyráběnou řadu. Na obrázku je znázorněno blokové schéma mikropočítače HC11. [2, 13]



Obr. 1 Blokové schéma HC11

1.3.1 Součásti HC11

Paměť a porty

Paměti mikropočítače HC11 mají velikost ROM 8kB, RAM 256B a 512B elektricky mazatelná paměť EEPROM. Další součástí jsou komunikační kanály, označované jako porty. *Port A* je vstupně/výstupní. Zpřístupňuje systém časovačů. Mezi jeho funkce patří klasický čítač událostí (impulsní akumulátor), trojnásobná funkce zachycení vstupu (input capture), která umožňuje díky časové základně přesně zachytit okamžik změny logické hodnoty signálu a tuto změnu signalizovat vyvoláním přerušení. Přes tento port je k mikropočítači připojena klávesnice. *Port B* je výstupní portem, má výstupní adresy A8 - A15 pro adresování vnější paměti. *Port C* je vstupně/výstupním komunikačním kanálem. Tyto tři porty (A, B, C) zajišťují paralelní komunikaci. [1, 2, 12, 13]

Sériovou komunikaci zajišťuje *port D*. Šest vývodů portu D lze u mikropočítače HC11 přiřadit dvěma sériovým komunikačním kanálům nazývaným SCI (sériově asynchronní kanál) a SPI (sériově synchronní kanál). Kanál SCI slouží k příjmu a vysílání signálu. Kanál SPI je využíván k řízení periferních obvodů a řízení obvodu reálného času. *Port E* je vstupně výstupní a jeho funkcí je zpracování analogových a číslicových vstupů. [1, 2, 12, 13]

Registry

Další součástí jsou registry, označovány také jako akumulátory. Mohou být osmibitové, do této skupiny patří registry A a B, které mohou být spojeny do šestnáctibitového registru D, kde registr A tvoří horní byte a registr B dolní byte. Je nutné si uvědomit, že načtením dat do registru D automaticky ztratíme původní hodnoty registrů A a B. Dále je k dispozici také osmibitový příznakový registr (CCR). Zbývající registry jsou šestnáctibitové. Do této skupiny patří indexové registry označované IX, IY, které slouží jako pracovní registry pro dočasné uložení dat. Dalším registrem je registr SP (Stack Pointer) ukazatel na vrchol zásobník. Poslední registr je označován PC (Program Counter) neboli čítač instrukcí, který není přímo programově přístupný. Po resetu je tento registr nastaven na nulu. Ukazuje vždy adresu, ze které se bude číst následující instrukce. Po jejím načtení z paměti se zvýší (inkrementuje) o jedničku. [1]

1.3.2 Systém přerušení

Přerušovací systém zajišťuje rychlou reakci mikroprocesoru a mikropočítače na nějaký podnět z okolí mikropočítače, např. odpovídá na žádosti vstupně/výstupních zařízení nebo na žádosti přímo z technického vybavení mikropočítače či mikroprocesoru. Téměř naprostá většina mikroprocesorů je vybavena možností tzv. přerušení. Tato možnost, podporována hardwarovým uspořádáním mikroprocesoru, je pak v mikropočítačovém systému rozšířená o celý systém přerušení. Tento systém pak obsahuje ve své struktuře několik vstupů žádosti o přerušení. Na tyto vstupy mohou být připojeny zdroje přerušení, které aktivují žádost o přerušení v libovolných okamžicích. Systém provede výběr žádostí, které mají v daný okamžik význam. Ostatní mohou být zakázány. V mikropočítačové terminologii se tomuto výběru říká *maskování zdrojů přerušení*. Dále se řeší priorita, to znamená v případě více žádostí o přerušení musí rozhodnout, která žádost bude obsloužena jako první. V případě přijetí žádosti o přerušení následuje fáze identifikace zdroje. Přerušení během, kterého se začne vykonávat obslužný program přerušení. Dojde k přerušení činnosti původního programu a chystá se k předání řízení jinam. Musí se také uložit informace, jež nám později umožní po vykonání obslužného programu přerušení, vrátit se k provádění původního programu. Ještě před provedení první instrukce obslužného programu je nutné uložit obsah programového čítače, který určuje návratovou adresu. Navíc jsou také uschovány obsahy dalších registrů. Toto již však provádíme obslužnými instrukcemi v obslužném programu přerušení. [1, 2, 12, 13]

Přerušovací vektory jsou všechny umístěny před koncem adresovatelného prostoru. Tento prostor je zpravidla vyplněn pamětí ROM (EPROM). Vektory přerušení představují standardní adresy, na kterých se nachází opět ve většině případů standardizované adresy, směřující do oblasti vnitřní paměti RAM. Přehled všech přerušení je dán tabulkou. Priorita uvedených přerušení je dána pořadím jejich uvedením v tabulce [1].

Máme dva druhy přerušení. Pokud existuje více žádostí o přerušení, mohou být obsluhovány postupně a to tak, že žádost později aktivovaná je obsloužena až po skončení obslužného programu předchozího přerušení. Tomuto zpracování říkáme *jednoúrovňové přerušení*. Druhou možností je přerušit i obslužný program přerušení. Tato situace může nastat v případě dalšího výskytu žádosti o přerušení s vyšší prioritou. Nastává pak přerušení v přerušení neboli *víceúrovňové přerušení*.

Postup obsluhy přerušení

Požadavkem na přerušení se ohlásí některý z periferních obvodů mikrořadiče (vnějších nebo vnitřních). Pokud je přerušení povoleno (instrukcí CLI), centrální procesorová jednotka akceptuje požadavek na přerušení na konci právě vykonávané instrukce a současně se znemožní přijetí další žádosti o přerušení. Při nemaskovatelném přerušení XIRQ nastaví bity podmínkového registru X a I na 1, u ostatních přerušení pak nastaví pouze I na 1. Potom dojde k přemístění všech obsahů registrů (s výjimkou SP) do zásobníku a CPU přejde na provádění obslužného programu přerušení od adresy, která je určena příslušným vektorem přerušení. Po provedení programu obsluhy přerušení (po instrukci *RTI - Return from Interrupt*) se obnoví původní obsah pracovních registrů. Nejdůležitějšími druhy přerušení jsou maskovatelné, nemaskovatelné a reset [1, 2, 12].

Maskovatelné přerušení (IRQ)

Postup obsluhy tohoto přerušení je stejný jak bylo popsáno výše. Adresa obsluhy přerušení se získává z paměťového místa \$FFF2 a \$FFF3. Masku přerušení se nastaví na 1. Nastavením masky je definován postup v případě více současně vydaných požadavků na přerušení. Pokud je potřebné umožnit vnoření dalšího přerušení do obslužného programu, musí se maska přerušení, instrukcí CLI na začátku obslužného programu přerušení, vynulovat. Jestliže vnoření není požadováno, maska se nenuluje. Provedením instrukce RTI, která ukončí obsluhu přerušení, se ze zásobníku obnoví všechny registry. Před přerušením byla maska přerušení ve stavu 0, po provedení instrukce RTI je přerušení znovu povoleno. [1]

Nemaskovatelné přerušení (XIRQ)

Nemaskované přerušení se automaticky zamaskuje pouze po resetu maskovacím bitem X, v příznakovém registru (CCR). Po povolení přerušení ze vstupu XIRQ v průběhu vykonávání programu nelze již tento maskovací bit vrátit zpět. Na XIRQ připojujeme obvykle obvody, které generují přerušení s nejvyšší prioritou. Nastaví se bitové masky X a I na 1. Vektor přerušení se získává z paměťových míst \$FFF4 a \$FFF5. [1]

Reset

Slouží k nastavení mikroprocesoru do počátečního stavu . CPU nastaví masky přerušení (I a X) a získává vektor přerušení z paměťových míst \$FFFE a \$FFFF.

Reset mikrořadiče se může uskutečnit také interně signálem COP (nazývaným též Watch dog) [1, 2].

1.3.3 Programování HC11

Zdrojové programy pro překladač se zapisují v jazyce symbolických adres (instrukcí). Jazyk symbolických adres je nejnižším programovacím jazykem prakticky používaným při programování počítačů a mikropočítačů. Z hlediska rozkladu úlohy odpovídá programování v tomto jazyce programování ve strojovém jazyce. Každé strojové instrukci je přiřazena instrukce jazyka symbolických adres. Jazyk symbolických adres je tedy strojově závislým jazykem, neboť programátor se musí seznámit se všemi vlastnostmi počítače na úrovni jeho strojových instrukcí. Instrukce jsou však na rozdíl od strojového jazyka zapisovány symbolickou formou. Symboly jsou tvořeny většinou mnemotechnickými zkratkami slov přirozeného jazyka (především angličtiny), zápis programu je tak čitelnější a srozumitelnější než zápis ve strojovém jazyce.

Překlad programu z jazyka symbolických adres do strojového jazyka zajišťuje překladač tzv. assembler. Díky strojovému překladu lze změny programu, jeho přemístění v operační paměti počítače, vyřešit podstatně jednodušeji než ve strojovém jazyce. Programy v jazyce symbolických adres mají obvykle menší nároky na rozsah operační paměti než programy zapsané ve vyšších programovacích jazycích. Také jejich optimalizace z hlediska doby potřebné pro výpočet programu je snazší. Nevýhodou jazyka symbolických adres je nemožnost přenést programy na jiný počítač, pracnost programování a omezené možnosti syntaktických a sémantických kontrol programů při překladu i výpočtu. [1, 2, 3, 12]

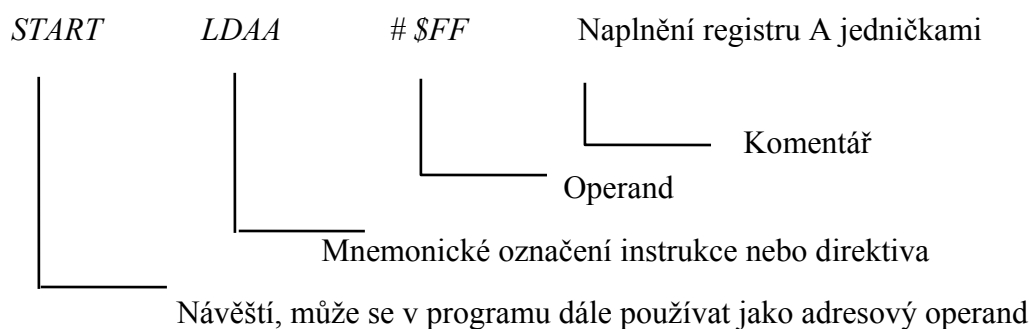
Zápis instrukcí v assembleru

Důležitou podmínkou vykonávání každého programu je správné umístění instrukcí a operandů v paměti tak, aby mikroprocesor při jejich čtení a vykonávání nezabloudil. Způsob, jakým jsou navzájem přiřazeny instrukce a jejich příslušné operandy se nazývá způsob adresování. [1] Struktura instrukčního slova prováděné v jednom instrukčním cyklu procesoru je zobrazeno na obrázku .

OPERAČNÍ KÓD	ZPŮSOB ADRESOVÁNÍ	POLE ADRESY
--------------	-------------------	-------------

Obr. 2 Struktura instrukčního slova

Zdrojový jazyk připouští použití množiny znaků, které assembler umí rozpoznat. Patří sem písmena anglické abecedy, číslice 0 až 9, znaky používané jako speciální prefixy (# - určuje bezprostřední adresování, \$ - šestnáctkové (hexadecimální) číslo, % - dvojkové číslo, apostrof - určuje znak ASCII). Také umí rozpoznat speciální znaky aritmetických operací (+, -, *, /), čtyři speciální znaky jako oddělovače (mezera, horizontální posuv - TAB, CR - konec řádku a čárku). Ve zdrojovém řádku za instrukčním slovem může následovat komentář, který může obsahovat znaky ASCII, ale nepřekládá se. Zdrojový program, který vytváříme textovým editorem, se skládá z posloupnosti zdrojových řádků. Zdrojový řádek lze zpravidla rozdělit do čtyř polí např. v následujícím uspořádání viz. Obr. 3.



Obr. 3 Zdrojový řádek

Protože návěští určuje adresu instrukce, není dovolena dvojznačnost ve specifikaci hodnoty návěští. Návěští je symbolické jméno. Jménu je při překladu přiřazena konkrétní adresa (symbolická adresa je nahrazena adresou skutečnou). Dané návěští smí být deklarováno pouze jednou. Při vícenásobném výskytu téhož jména v poli návěští hlásí překladač chybu. Při vytváření symbolických jmen nelze použít vyhrazené symboly překladače, tedy názvy instrukcí a pseudoinstrukcí, stejně tak názvy registrů. [1, 2, 3, 17]

Instrukční soubor

Přehled všech instrukcí, které lze použít při programování. Instrukce jsou rozděleny do několika základních skupin [1, 2, 12, 17].

1. Instrukce pro aritmetické operace (přičtení a odečtení obsahu paměťových buněk, inkrementace paměťových buněk atd.).
2. Instrukce pro logické operace (např. logický součin, součet, aritmetický posun bitů)
3. Řídící instrukce (čekání na přerušení, softwarové přerušení)
4. Instrukce pro práci se zásobníkem (např. ukládání dat mezi zásobníkem a akumulátorem)
5. Instrukce pro řízení toku programu (nepodmíněné a podmíněné skoky atd.)
6. Instrukce pro nastavení počátečního stavu (vynulování obsahu paměť. buňky, ...)
7. Instrukce pro přenos dat mezi CPU a pamětí (uložení hodnoty z operandu do akumulátoru)

1.4 Operační systémy pro řízení v reálném čase

Systémy pracující v reálném čase jsou takové systémy, které vždy musí dát odpověď na vnější signál v konečné, předem stanovené, lhůtě. Jsou tedy předem stanoveny maximální doby zpoždění reakce na dané podněty. Tyto systémy jsou víceuživatelské, kdy více uživatelů může pracovat na jednom počítači současně a víceúlohové, umožňují práci v režimu sdílení času, kdy je v rámci multitaskingu provozováno více úloh patřících různým uživatelům. Multitasking tedy umožňuje souběžné zpracování více úloh na jednom počítači, kdy jedna úloha probíhá na popředí (foreground) a ostatní probíhají na pozadí (background). Dovoluje lepší využití CPU. V případě, že uživatel pracuje interaktivně s nějakým programem, který většinu času čeká na zadání jeho požadavků, je možné, aby procesor prováděl např. nějaký náročný matematický výpočet. Je-li na počítači s jedním procesorem provozováno více programů, je procesor přidělován postupně vždy na určitou dobu, tzv. časové kvantum (asi 0.1 s). Podle způsobu práce rozlišujeme dva druhy multitaskingu [1, 2, 7].

Kooperativní multitasking

Procesor je operačním systémem přidělen jednomu programu, který jej má v držení tak dlouho, dokud jej sám nevrátí zpět operačnímu systému. Ten jej pak přidělí jinému programu. Nevýhodou je, že program nemusí procesor navrátit v dostatečně krátkém časovém úseku, což způsobí dojem, že ostatní programy nepracují. Ještě horší případ nastane ve chvíli, kdy program procesor nevrátí vůbec (např. zhavaruje). Tato situace vede ve většině případů k havárii celého systému. [7]

Preemptivní multitasking

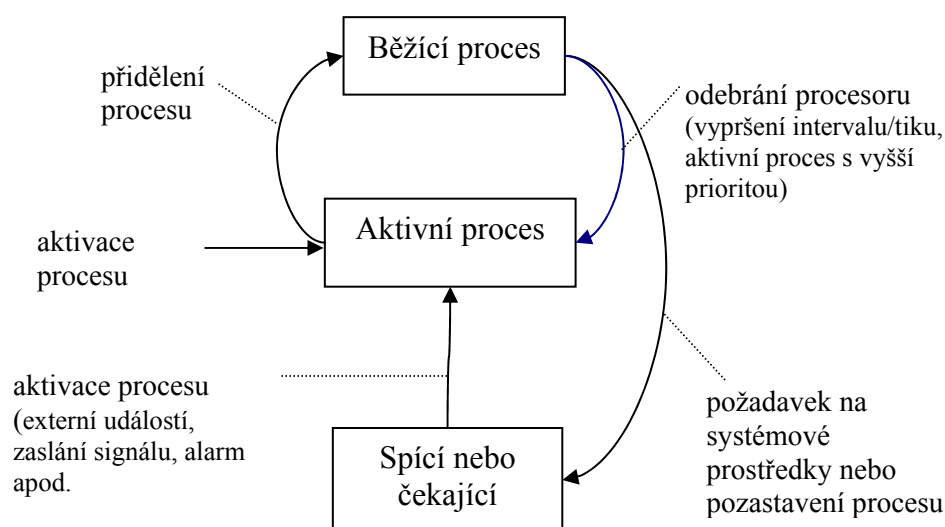
Procesor je přidělen programu pouze na určitou dobu a po jejím uplynutí jej sám operační systém programu odebere a přidělí jinému programu. Proces může být přerušen plánovačem kdykoliv. Z toho vyplývá, že nemohou nastat stavy uvedené u kooperativního multitaskingu. Nevýhodou tohoto řešení je vyšší náročnost na hardwarové vybavení počítače. [7]

1.4.1 Stavy procesů

Procesy se mohou dostat do tří základních stavů. Tyto stavy se týkají jen doby zpracování procesu, tedy od doby spuštění až do jeho ukončení. První stavem procesu je stav *aktivní – připraven* (active – ready). Počet procesů nacházejících se v tomto stavu závisí na parametrech operačního systému a daném počítači. Tato skupina bývá označována jako aktivní fronta procesů nebo také fronta připravených procesů na zpracování. Proces je zde zařazen pokud je nějakým způsobem spuštěn – aktivován nebo pokud je přerušeno jeho zpracování operačním systémem. [1, 7]

Druhým stavem je stav *běžící – zpracováváný* (running - execute). Běžící proces je ten proces, který v daném okamžiku využívá čas procesoru. U víceprocesorových systému počet těchto procesů odpovídá počtu procesorů.

Třetím stavem procesu je stav *čekající* (waiting) nebo *spící* (sleeping). Proces buď čeká na nějakou událost, která má v operačním systému nastat, nebo je pozastaven - „uspán“. Počet procesů závisí na jejich velikosti a na velikosti operační paměti. V tomto stavu proces nežádá o přidělení procesoru, ani není začleněn do plánovacích struktur. Je pasivně uložen v operační paměti a může být aktivován operačním systémem. Tyto stavy jsou zaznamenány na Obr. 4.



Obr. 4 Průběh vykonávání procesů

1.4.2 Plánování procesů

Při realizaci víceúlohového zpracování jsou pro výměnu procesů používány různé strategie – postupné plánování, plánování s cyklickou obsluhou, prioritní plánování a víceúrovňové strategie. Pro zpracování procesů o stejné prioritě se používá cyklické plánování. Je nastaven časový interval – kvantum, po jehož uplynutí se zpracování příslušného procesu přeruší a začne se zpracovávat z příslušné fronty. Postupně jsou obslouženy všechny procesy o stejné prioritě.

Nejčastěji se zpracování řídí podle dynamické priority. Priorita čekajících procesů je zvyšována ve zvoleném časovém intervalu. Při vstupu do fronty připravených procesů mají stejnou prioritu, která se s dobou čekání pravidelně zvyšuje. U další metody se priorita procesu mění podle délky zpracování procesu. Proces s nejkratší dobou zpracování má nejvyšší prioritu. Její nevýhodou je, že většinou nevíme jak dlouho se bude proces zpracovávat. Je však vhodný pro řízení v reálném čase, kde délka zpracování procesu je známa. U kombinované metody se dynamická priorita určuje v závislosti na poměru doby odezvy a doby zpracování. [7]

1.4.3 Průběh vykonání procesu

V neaktivním stavu jsou úlohy uloženy v souborech v systémové nebo uživatelské oblasti diskové paměti. Neaktivní úlohy mohou být převedeny do aktivního stavu operátorským požadavkem, dosažením stavu, ve kterém je úloha naplánována ke spuštění, systémovým požadavkem a externí událostí. Aktivní procesy se řadí do fronty procesů

připravených na zpracování. Fronta je nejčastěji typu FIFO (First-In First-Out). [12] Frontu požadavků na zpracování procesů obsluhuje plánovač – dispečer (scheduler). Jeden z procesů pro zprávu prací operačního systému. Z aktivních procesů se vybírá podle zvolené strategie příslušný proces, kterému se přidělí procesor a zahájí se jeho zpracování. Proces se může dostat do několika stavů čekání: čekání na procesor, na I/O zařízení, na dokončení procesu vázaného generickým vztahem (např. „rodič-potomek“), na obecnou událost, čekání na systémové prostředky nebo může být pozastaven operátorem nebo jiným procesem – spící a čekající na obnovu aktivace. [1, 7]

Procesy aktivované v závislosti na čase, jsou trvale zařazeny v seznamu úloh spouštěných v závislosti na čase. Tento seznam obsluhuje jeden ze systémových procesů operačního systému, který nastavuje hodnotu systémového času ve spolupráci s časovačem. Prohlíží seznam v čase spouštěných procesů a je-li plánovaný čas zpracování shodný se systémovým časem, provede aktivaci příslušných procesů. Při nedostatku místa pro čekající procesy v operační paměti jsou procesy odkládány na disk. Odkládání lze pomocí systémových prostředků potlačit. Využití při zpracování úloh v reálném čase. [1, 7]

1.4.4 Reálný čas a mikropočítače

U osmibitových mikropočítačových systémů se jedná o operační systém, který umožňuje multitasking, přičemž je využíván jeden zdroj přerušení – přerušení od časovače. Je zpracován ve formě modulů, které se linkují společně s moduly, realizujícími jednotlivé procesy. Celý programový systém včetně RTMONu tvoří jeden exe soubor. Vzniká omezení – nelze zavádět dynamicky jednotlivé moduly do operační paměti. Všechny moduly, které využívají služby RTMONu musí obsahovat hlavičkový soubor `#include <rtm.h>`. Pro inicializaci musí být v hlavním programu volána služba `init_rtm()`. V tomto okamžiku se vytváří proces `init` s prioritou 0 a proces `dummy` s prioritou 255, který vykonává dynamickou smyčku, pokud žádný proces není aktivní. Při vytvoření dalšího procesu službou `rtm_create_p()`, musí být inicializační proces pozastaven službou `rtm_delay_p()`. Předáno řízení spuštěného uživatelského procesu s nejvyšší prioritou, spuštění procesu službou `rtm_start_p()`, nastavujeme periodu, se kterou je proces spouštěn. Toto musíme udělat, protože inicializační proces s nejvyšší prioritou by nepustil žádný jiný proces. Dále program pokračuje procesem, který jsme spustili.

Při ukončení činnosti musí být spuštěn hlavní program, znovuspuštěním inicializačního procesu, voláním `rtm_continue_p()`. Musí být zastaveny `rtm_stop_p()` a abortovány (zrušeny) `rtm_abort_p()` všechny aktivní či existující procesy. U periodicky spouštěných procesů se musí nejprve zrušit periodické spouštění tj. změnit perioda spouštění na jednorázové. Na konec je nutné zavolat služba `rtm_end()`, která ukončí činnost procesů `init` a `dummy`. Zbytek programu doběhne jako běžný DOSový program, tedy pokračujeme bez reálného času. V tabulce je uveden přehled a stručný popis služeb systému pro řízení běhu procesu. [1]

Tabulka I Přehled služeb systému pro řízení běhu procesu

Označení služby	Provedená akce
<code>int rtm_init (IDPROC ** init_id);</code>	Inicializace jádra
<code>int rtm_end (IDPROC * init_id);</code>	Ukončení práce jádra
<code>int rtm_create_p(char * pname, char prio, void(*pfunc)(),int stack_size, IDPROC** proc_id);</code>	Vytvoření procesu - alokují se datové struktury, zásobník
<code>int rtm_start_p(IDPROC * proc_id, long time_to_start, long time_period);</code>	Spuštění procesu - proces musí být ukončen <code>rtm_stop_p</code> , aby mohl být opět spuštěn
<code>int rtm_delay_p(IDPROC * proc_id, long time_to_delay);</code>	Pozastavení procesu - je-li pozastaven na neurčitou dobu, může se znovuspustit službou <code>rtm_continue_p</code>
<code>int rtm_continue_p(IDPROC * proc_id);</code>	Znovuspuštění pozastaveného procesu
<code>int rtm_ch_period_p(IDPROC * proc_id, long time_period);</code>	Změna periody volání
<code>int rtm_abort_p(IDPROC * proc_id);</code>	Abortování (zrušení) procesu - zruší se celý proces včetně jeho datových struktur
<code>int rtm_stop_p(IDPROC * proc_id);</code>	Zastavení procesu - proces je okamžitě zastaven

Legenda:

init_id	- identifikátor inicializačního procesu
* pname	- pointer na jméno procesu (max 12 znaků)
prio	- priorita procesu (1 - 254) nejvyšší je 1
* pfunc	- pointer na funkci void pfunc (void), která představuje vlastní proces
stack_size	- velikost zásobníku
* proc_id	- identifikátor procesu pro další ovládání procesu
time_to_delay	- časový interval (v ticích RTMON), na který má být proces pozastaven

2 TESTOVACÍ PROGRAMY

Úkolem bakalářské práce je připravit podklady pro testovací program. V současnosti jsou testy provozovány jako aplikace. Tyto aplikace testy vygenerují podle zadaných kritérií, vyhodnotí odpovědi studenta a výsledky testů uloží do databáze. Při nastavení testu chceme zadat větší množství otázek. Po aplikaci pak vyžadujeme, aby nám z nich náhodně vygenerovala zadaný počet otázek a výsledky testů. Dalšími méně důležitými nároky na aplikaci je časové omezení na zvládnutí testu, popř. hodnocení studenta. Porovnáváme dva již vytvořené programy pro testování znalostí studentů. První z nich byl vytvořen jako diplomová práce a je k dispozici na školním serveru <http://testy.utb.cz>. Testy můžeme také vytvářet v databázi Moodle, která je na serveru <http://vyuka.ft.utb.cz>.

V této části bakalářské práce jsou popsány již dříve zmíněné aplikace. Jejich porovnáním je rozhodnuto, která z nich lépe splňuje námi zadané požadavky.

2.1 Testy utb

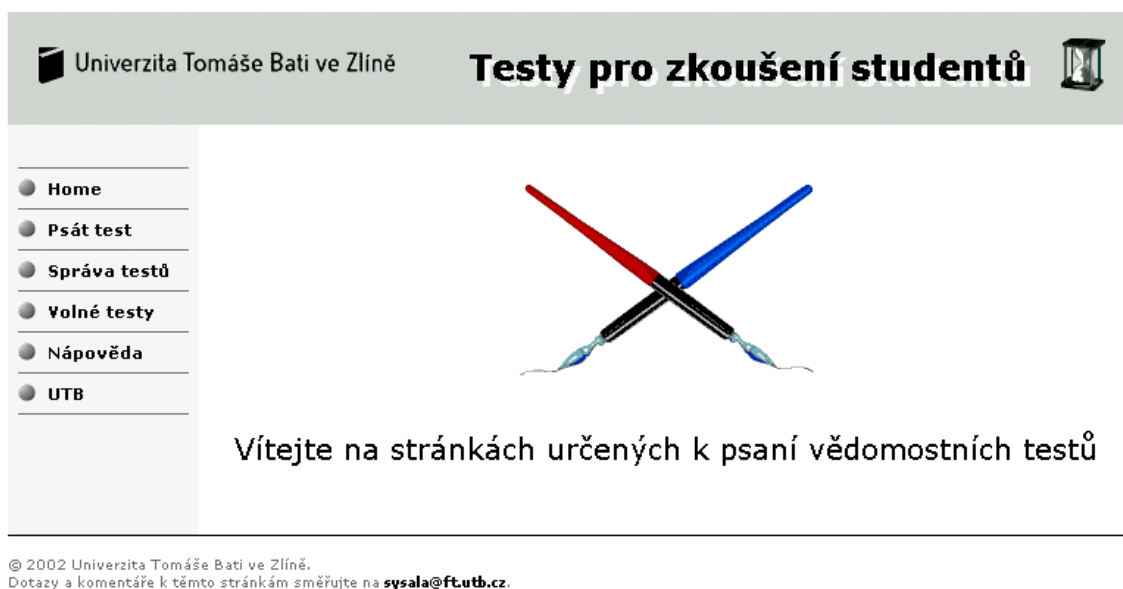
2.1.1 Aplikace TESTY

Aplikace testy je k dispozici na školním serveru <http://testy.utb.cz>. Tato aplikace je vytvořena ve skriptovacím jazyce PHP a pracuje s databází MySQL. WWW stránky jsou napsány pomocí jednoduchých příkazů HTML s využitím technologie CSS, umožňující definovat grafickou podobu a chování jednotlivých prvků stránky. Formulářů je využito k interaktivní komunikaci mezi klientem a aplikací. Pomocí formulářů se vytváří testy, přihlašuje se do systému a psaní testů. Aplikace je rozdělena do tří částí. První částí je *Admin*, umožňující správu aplikace. Druhou částí *Správa testů*, slouží k vytváření nových testů, úpravě stávajících testů, jejich mazání a přístup k výsledkům testů. Poslední částí je *generátor testů*, který testy generuje a opravuje. [8]

Každý uživatel je autorizován jménem a heslem. Pokud má uživatel nastavena práva k přístupu do administrátorské části, je do této části přesměrován logovacím skriptem. Má-li nastaven práva jen pro správu testů, je přesměrován do části Správa testů.

2.1.2 Úvodní stránka

V hlavní nabídce si můžeme zvolit jednu z těchto možností - *Home*, *Psát test*, *Správa testů*, *Volné testy*, *Nápověda*, *Utb*. Stisknutelné možnosti jsou v této aplikaci označeny tučným písmem.



Obr. 5 Úvodní stránka systému testy

Home

Přechod na úvodní stránku Webu.

UTB

Odkaz na Univerzitu Tomáše Baťi ve Zlíně.

2.1.3 Psát test

Test: MIKROPOČÍTAČE

Jedinou správnou odpověď označte **TEČKOU** v **ODPOVĚDÍCH**.

Test vytvořil: **NOVOSADOVA**

Jméno zkoušeného:	Ročník studia:	Studijní skupina:	Začátek testu:	Dnešní datum:	IP adresa:
	1. 		13:57:16	4.05.2005	212.80.64.66

1. Jak se jmenuje garant předmětu?

- ☐ a) profesor Vašek
- ☐ b) ing. Dostálek
- ☐ c) ing. Dolinay
- ☐ d) ing. Bližňák

2. Jakou zkratkou se označuje příznakový registr ?

- ☐ a) CCR
- ☐ b) CCB

Obr. 6 Psát test

Zvolíme-li si položku *Psát test* objeví se nám tabulka, do které musíme zadat *jméno* testu a *přístupové heslo*. Pokud jsou vyplněny platné údaje, systém nám povolí pokusit se o zvládnutí testu. V opačném případě, se objeví upozornění, že jsou chybně zadány údaje pro přihlášení. Volbou zpět se můžeme pokusit o přihlášení znovu.

Test se nám nepovede odeslat v případě, že jsme nezadali *jméno zkoušeného* nebo *číslo skupiny*. Dále je zaznamenán čas začátku testu, společně s datumem a IP adresou. V testu pak vybíráme z nabídnutých odpovědí, přičemž jen jedna z nich je správná. Pokud jsme odpověděli na všechny otázky, potvrdíme nabídku *Odeslat test*.

Opraven test: **mikropočítače**

Jméno zkoušeného:	Peterka Jiří
Ročník:	2
Studijní skupina:	12
Začátek testu:	13:57:16
Konec testu:	14:02:02
Dnešní datum:	4.05.2005

Odpovědi:

Otázka č.	Odpověď
1	správně
2	správně
3	správně
4	správně
5	správně
6	správně
7	správně
8	špatně
9	správně
10	bez odpovědi
11	správně

Hodnocení:

Dosažený počet bodů:	9/13 , 70%/100%
Klasifikace:	nevyhověl

Výsledky zapsány do databáze.

Odejít

Obr. 7 Psát test – odevzdání testu

Po odeslání testu se zobrazí stránka, na které jsou uvedeny všechny údaje, které jsme zadávali při provádění testu, spolu se seznamem odpovědí, kde je u čísla otázky zaznamenáno, zda jsme ji zodpověděli správně či nikoli. Také se zobrazuje dosažený počet bodů, které jsme v testu získali ve formátu : počet získaných bodů / ze všech možných a jeho procentuální úspěšnost ve formátu : počet procent získaných v testu / počet procent všech možných. Klasifikaci určuje správce daného testu. Klasifikace je rozdělena do čtyř skupin. Pokud student v testu prospěl, může získat známku 1, 2 nebo 3. Nebyly-li jeho odpovědi dostatečné, získá hodnocení nevyhověl. Výsledky studentů jsou odeslány do databáze, ke které má přístup daný správce testu. Volbou možnosti *Odejít* se dostaneme na úvodní stránku Webu.

2.1.4 Správa testů



Obr. 8 Správa testů - úvodní stránka

Při označení položky *Správa testů* je zobrazena tabulka, do které je zadáváno jméno správce a jeho přístupové heslo. Pokud jsme vyplnily platné údaje, dostaneme se do správy testu daného uživatele. V opačném případě se nám vypíše varování, že některý z údajů byl chybně zadán.

V horní části pod názvem *Správa testů* je zobrazena nabídka, ze které má správce možnost vybírat. Jsou zde tyto možnosti *Nastavení uživatele*, *Testy*, *Založit test*, *Výsledky testů*, *Nápověda*, *Odhlásit*. Tyto volby jsou jednotlivě popsány v následujícím textu.

Nastavení uživatele

Správa testů

[Nastavení uživatele](#) [Testy](#) [Založit test](#) [Výsledky testů](#) [Nápověda](#) [Odhlásit](#) **Přihlášen: NOVOSADOVA**

Nastavení uživatele

Jméno uživatele Heslo* e_mail

NOVOSADOVA

Poznámka: Údaje označené * musí být vyplněny.

© 2002 Univerzita Tomáše Bati ve Zlíně.
Dotazy a komentáře k těmto stránkám směřujte na sysala@ft.utb.cz.

Obr. 9 Správa testů – nastavení uživatele

Vypíše se jméno příslušného správce. Zde má správce možnost změnit své heslo, popřípadě e-mail, který je také zobrazen.

Testy

Správa testů

[Nastavení uživatele](#) **Testy** [Založit test](#) [Výsledky testů](#) [Nápověda](#) [Odhlásit](#) **Přihlášen: NOVOSADOVA**

Jméno testu Heslo testu Klasifikace 1 Klasifikace 2 Klasifikace 3

Jméno testu Jméno uživatele

Výpis testů: **aktualizuj výpis**

	Test	Heslo	Otázek	Odpovědí	Klasif. 1	Klasif. 2	Klasif. 3	Stav testu			
1.	konec	šifrováno	40	3	30	20	10	dokončen	edit	uvolnit	smazat
2.	mikro	šifrováno	40	3	37	30	20	nedokončen	edit	uvolnit	smazat
3.	mikropočítače	šifrováno	11	4	11	10	9	dokončen	edit	uvolnit	smazat

© 2002 Univerzita Tomáše Bati ve Zlíně.
Dotazy a komentáře k těmto stránkám směřujte na sysala@ft.utb.cz.

Obr. 10 Správa testů – výpis testů uživatele

Volbou možnosti *Testy*, se zobrazí kompletní výpis všech testů. Tento výpis je formou tabulky. V prvním sloupci jsou vypsaná jména testů. Potvrdíme-li některý test, zobrazí se náhled na test ve formě otázek a nabízených možností. Pokud v testu nejsou vytvořeny žádné otázky, vypíše se text „Test nemá vloženy žádné otázky“.

Ve druhém sloupci, který je nazván *Heslo*, se mohou objevit dvě možnosti. V případě, že test je zpřístupněn pouze po zadání hesla, vypíše se šifrováno. V opačném případě je test dostupný všem, čehož dosáhneme při volbě položky *Uvolnit*, která je zobrazena za tabulkou a bude popsána v následujícím textu. Tímto je test označen jako volný.

Další sloupec, nazván *Otázek*, je počet otázek, které test obsahuje. V následující sloupci tabulky, označeném jako *Odpovědi*, se vypíše počet odpovědí u každé otázky.

V následující třech sloupcích je zadána klasifikační stupnice. Ve sloupci s danou klasifikací je zaznamenán minimální počet bodů, který student musí získat, aby byl jeho test vyhodnocen danou klasifikací.

V posledním sloupci je stav testu. Je-li test dokončen, tedy musí být zadány všechny položky testu, je test označen jako dokončený, v opačném případě se vypíše nedokončený.

Na konci tabulky jsou u každého testu ještě další tři volby - *Edit*, *Uvolnit*, *Smazat*. Před tím, než tyto volby použijeme, je nutné zvolit možnost *Založit test*. Tato volba je popsána v následující části nazvané *Založit test*.

Jakmile je test vytvořen, po tom co jsme nastavily jeho parametry, tedy jméno, přístupové heslo, počet otázek, počet možností odpovědi u každé otázky a je zadána klasifikační stupnice, můžeme začít vytvářet otázky. V tabulce výpisu testů, v řádku se jménem testu, který chceme upravovat, zvolíme možnost *Edit*.

Správa testů

Nastavení uživatele
Testy
Založit test
Výsledky testů
Nápověda
Odhlásit
Přihlášen: NOVOSADOVA

Jméno testu: **MIKROPOČÍTAČE**

Otázek: 11/11

Otázka č.	Otázka	Body	
1	Jak se jmenuje garant předmětu?	1	upravit
2	Jakou zkratkou se označuje příznakový registr ?	1	upravit
3	Zásobníková paměť je typu	1	upravit
4	Z čeho se skládá mikroprocesor?	1	upravit
5	Co je na obrázku? ccr.bmp	1	upravit
6	Které registry jsou 8-bitové	1	upravit
7	Jaké má HC 11 porty?	1	upravit
8	Které porty HC11 jsou vstupně - výstupní?	2	upravit
9	Je HC11 jednočipový mikropočítač?	1	upravit
10	Jaké přerušení existuje	2	upravit
11	Kolik nohou má pes	1	upravit

© 2002 Univerzita Tomáše Bati ve Zlíně.
Dotazy a komentáře k těmto stránkám směřujte na sysala@ft.utb.cz.

Obr. 11 Správa testu – editace

Po této akci se zobrazí stránka, na které je jméno testu, který editujeme. Počet vytvořených otázek / počet otázek, kolik jich chceme v testu mít – zadávali jsme při akci *Založit test*. Dále se zobrazuje tabulka s poli *Otázka číslo*, *otázka* - text otázky a *body* - počet bodů za správné zodpovězení otázky. Při prvním přístupu místo textu otázky se vypíše text „Test neobsahuje žádné otázky“. Je nám také nabídnuta akce *Vložit otázku*. Po jejím potvrzení se zobrazí nová stránka.

Správa testů

Nastavení uživatele
Testy
Založit test
Výsledky testů
Nápověda
Odhlásit
Přihlášen: NOVOSADOVA

Jméno testu: **MIKRO**

Otázka:

Vložit obrázek do otázky

Procházet...
Vlož obr. do otázky

Odpověď č.1

Vložit obrázek do odpovědi

Procházet...
Vlož obr. do odpovědi

Odpověď č.2

Vložit obrázek do odpovědi

Procházet...
Vlož obr. do odpovědi

Obr. 12 Správa testů – vkládání otázek

Na této stránce je opět zobrazeno jméno testu, se kterým pracujeme. Do prvního okna pod označením *Otázka*, zapíšeme text otázky. K otázce můžeme také vložit obrázek, zvolíme-li možnost *Procházet*, vedle okna kam jsme psali text otázky. Zobrazí se nové okno, nazvané *Zvolte soubor*, kam zadáváme umístění souboru, který chceme přidat, na disk. Obrázky, vkládané do testu, musí být ve formátu, který podporuje WWW prohlížeč, tedy soubory s příponami gif, jpg nebo png. Zadáme-li soubor, musíme zvolit *Vložit obrázek do otázky*.

Pod zadáváním otázky je zadávání odpovědí. Sem vpisujeme text odpovědi. Také k odpovědím můžeme přidávat obrázky. Postup je stejný jako při vkládání obrázku k otázce. Po nastavení otázky, se nám vždy nabízí všech 10 možných odpovědí, i když jsme při nastavení zadávali, že chceme jen např. čtyři. Pokud vyplníme více odpovědí než kolik jsme jich zadali při založení testu, budou se nám při vykonání testy odpovědi v náhodném pořadí střídát.

Máme-li zadánu otázku i odpovědi, ve spodní části stránky pak zadáme číslo správné odpovědi a počet bodů za otázku. Při zvolení volby *Reset* jsou vyplněné údaje vymazány. Zvolíme-li volbu *Uložit otázku*, zobrazí se stránka s textem : „Otázka uložena do databáze“. Volbou *Zpět* můžeme zadávat další otázku. Máme-li zadány všechny otázky, po uložení nastavení poslední otázky se vypíše „Databáze dokončena“. Zvolením možnosti *Testy* u jména vytvořeného testy je ve sloupci stav testu vypsáno dokončen. Kliknutím na název testu se zobrazí náhled na test.

Další možností na této stránce je volba *Uvolnit*. Lze ji použít pouze v případě, že již máme test dokončen – je vytvořeno tolik otázek, kolik jsme jich zadali při založení testu. Tato volba nám test uvolní. Budou k němu mít přístup všichni studenti bez omezení. Volné testy budou popsány v části *Volné testy*. Jakmile test označíme jako volný, nelze již toto nastavení změnit.

Poslední možností této stránky u výpisu testů je volba *Smazat*. Jak již vyplývá z názvu této položky je test z databáze navždy vymazán. Po zvolení této volby jsme dotázáni, zda chceme test opravdu smazat. Potvrzením je test nenávratně smazán.

Založit test

Správa testů

[Nastavení uživatele](#) [Testy](#) [Založit test](#) [Výsledky testů](#) [Nápověda](#) [Odhlásit](#) [Přihlášen: NOVOSADOVA](#)

Definice testu

Jméno testu:

Heslo testu:

Počet otázek:

(max. 40)

Počet odpovědí:

Klasifikace 1 od:

 bodů

Klasifikace 2 od:

 bodů

Klasifikace 3 od:

 bodů

Založ test

Vymaž

Obr. 13 Správa testů – založení testu

Zde se definují základní parametry testu. Všechny tyto parametry musí být vyplněny. Prvním parametrem je *Jméno* testu musí být vyplněno bez mezer. Test nebude založen pokud již v databázi test s tímto jménem existuje. Současně se prohledávají i názvy testů ostatních uživatelů, jméno testu tedy musí být unikátní v celé databázi.

Dále se zadává *Heslo* testu, které bude sloužit k přístupu k testu. Heslo musíme zadat také v případě, že chceme vytvořit test, ke kterému bude volný přístup. Test uvolníme volbou *Uvolnit test*, kterou jsem zmiňovala v předchozí kapitole. Dalším parametrem testu je *Počet otázek*, zadáváme kolik otázek bude test obsahovat. Počet otázek musí být zadáván v rozmezí 3 až 40 otázek, jinak by se nám nepovedlo test založit z důvodu nastavení klasifikace. U možnosti *Počet odpovědí* zadáváme kolik odpovědí se bude generovat pro jednu otázku. Otázka může mít minimálně 3 a maximálně 10 odpovědí. Také zadáváme *Klasifikační stupnici* podle bodů získaných v testu. Tato stupnice je rozdělena do čtyř skupin. Student může získat hodnocení výborný (1), velmi dobrý (2), dobrý (3). Klasifikační stupnice platí do uvedeného počtu bodů. V případě že se studentovi nepodaří získat minimální požadovaný počet bodů, je ohodnocen jako nevyhovující (4). Počet bodů u jednotlivých klasifikací se nesmí překrývat, proto musí být počet získaných bodů v testu nejméně tři.

Příklad:

Jméno testu – Mikropočítačel

Heslo testu - 12345

Počet otázek - 10

Počet odpovědí - 4

Klasifikace 1 - 20 bodů

Klasifikace 2 - 17 bodů

Klasifikace 3 - 12 bodů

Dle uvedených parametrů je založen test se jménem Mikropočítačel s přístupovým heslem pro psaní testu 12345. Otázek v testu je deset a u každé budou k dispozici čtyři odpovědi. Klasifikace je nastavena tak, že při získání minimálně 12 bodů je student hodnocen známkou *dobrý*, dosažením 17 bodů *velmi dobrý* a při 20 a více bodech *výborný*.

Po vyplnění potvrdíme možnost *Založ test*. V tomto okamžiku se nám vypíše parametry testu, tak jak jsme je zadali. Vkládání otázek bylo popsáno výše v části *Testy / editovat*. Volbou tlačítka *Vymaž* je obsah všech vyplněných buněk vymazán.

Výsledky testů

Správa testů

Nastavení uživatele

Testy

Založit test

Výsledky testů

Nápověda

Odhlásit

Přihlášen: NOVOSADOVA

Výsledky testů:

	Zkoušený	Test	Roč. studia	Stud. skupina	IP	Odpovědi	Bodů	Klasifikace	Zač. testu	Konec testu	Datum	
1.	Horká Verča	maja	2	11	212.80.64.66	11100	3	4	11:37:20	11:37:49	29.04.2005	smazat
2.	Ifog	sedmý	1	5	212.80.64.66	1110000	3	4	11:31:14	11:31:35	14.04.2005	smazat
3.	lokr	snaha	1	15	212.80.64.66	111001	4	4	11:12:12	11:12:29	14.04.2005	smazat
4.	Novotný Petr	maja	2	13	212.80.64.66	10011	3	4	11:35:00	11:36:14	29.04.2005	smazat
5.	Nový Hugo	maja	2	13	212.80.64.66	10111	4	4	11:35:00	11:35:52	29.04.2005	smazat
6.	Pokorný Pavel	maja	2	12	212.80.64.66	11111	5	1	11:35:00	11:35:31	29.04.2005	smazat
7.	Skácel Jiří	maja	2	15	212.80.64.66	1x101	3	4	11:38:00	11:38:28	29.04.2005	smazat
8.	Starý Pavel	maja	2	13	212.80.64.66	10010	2	4	11:35:00	11:36:39	29.04.2005	smazat

© 2002 Univerzita Tomáše Bati ve Zlíně.
Dotazy a komentáře k těmto stránkám směrujte na sysala@ft.utb.cz.

Obr. 14 Správa testů – výsledky testů

Na této stránce má uživatel k dispozici výsledky k testům, které vytvořil. *Výpis výsledků* je také rozdělen do tabulky. V prvním sloupci se vypisuje jméno zkoušeného podle toho, jak jej zadá při psaní testu. Dále je vypsáno jméno testu. Vypíše se také ročník a studijní skupina, tyto údaje zadává student. Dále se vypisuje IP adresa počítače, na kterém je test vykonán.

Je zobrazeno jak student odpovídal na jednotlivé otázky. Pořadí otázek je ve všech testech stejné. Odpovědi, které byly správné jsou označeny jedničkou, špatné nulou. Pokud na některou otázku neodpovídal je označena křížkem. Ve výpise se nám tedy v poli odpovědi vypíše kombinace jedniček, nul, popř. křížků. Další dva sloupce pak představuje bodové ohodnocení studenta a jeho klasifikaci. Klasifikace pouze číselně, slovní vyjádření zde chybí. Je také zaznamenána doba, jak dlouho student test psal, přesněji čas začátku a konce testu, včetně datumu.

Na konci tabulky u záznamu jednotlivých studentů je také volba *Smazat*. Tou můžeme postupně vymazat všechny záznamy. Pokud vymažeme nějaký test, jeho výsledky zůstanou v databázi do té doby, než je odtud vymažeme. Výsledky testů lze setřídít a to podle zkoušeného, jména testu, ročníku, studijní skupiny, bodů či podle klasifikace. Chtěla bych zmínit, že zde nejsou výsledky testů, které jsme uvolnili. Při psaní testu student nezadáva žádné své údaje.

Nápověda

Zde jsou stručně popsány jednotlivé položky hlavního menu, které byly popsány v předchozím textu, jako je *nastavení uživatele*, *testy*, *založit test*, *výsledky testů* a *odhlášení*.

Odhlásit

K odhlášení bych zmínila jen to, že je důležité se po ukončení práce ve Správě testů odhlásit, aby byla aplikace v korektním stavu. Po odhlášení přejdeme na úvodní stránku.

2.1.5 Volné testy

Volný test: ZNACKY

Jedinou správnou odpověď označte **TEČKOU** v **ODPOVĚDÍCH**.



1. Vyberte správný název dopravní značky

- ☐ a) Dej přednost v jízdě tramvaji !
- ☐ b) Dej přednost v jízdě !
- ☐ c) Přednost před protijedoucími vozidly



Obr. 15 Volné testy – psaní testu

V této části jsou uloženy testy, které jsme označili jako volné. Mají k nim přístup všichni studenti bez omezení. Tyto testy nejsou chráněny heslem. Jak již bylo zmíněno student zde nezadáva žádné své údaje, proto se výsledky testů nezaznamenávají v databázi. Slouží k tomu, aby si student mohl otestovat své znalosti sám pro sebe a zjistit jak tyto testy fungují. Po zvolení volných testů se objeví jejich seznam, kde je uvedeno jméno testu a autora. Po zvolení možnosti *Psát test*, student může začít psát test. Jakmile odpoví na otázky a zvolí *Odevzdat test*, zobrazí se tabulka. V tabulce je uveden datum, čas započetí a ukončení testu. Dále seznam čísel otázek, u každé je uvedeno, jak na které otázky odpovídal – správně, špatně nebo bez odpovědi. A v poslední řadě jeho hodnocení ve formě kolik získal bodů, jeho procentuální úspěšnost a klasifikace. Ukázka je na Obr.16. Pokud správce vymaže test z databáze, automaticky se tento test vymaže také z volných testů. Stisknutím volby Odejít se dostaneme na úvodní stránky Správy testů.

Opraven test: **ZNACKY**

Autor testu: **JIRI VESELY**
Začátek testu: **19:22:36**
Konec testu: **19:23:16**
Dnešní datum: **4.05.2005**

Odpovědi:

Otázka č.	Odpověď
1	špatně
2	správně
3	správně
4	správně
5	správně
6	správně
7	správně

Hodnocení:

Dosažený počet bodů: **30/35 , 86%/100%**
Klasifikace: **výborně**

Odejít

Obr. 16 Volný test – vyhodnocení testu

2.2 Testy Moodle

2.2.1 Úvod k Moodle

Moodle je softwarový balíček pro tvorbu výukových systémů a elektronických kurzů na internetu. Jedná se o neustále se vyvíjející projekt, navržený na základě sociálně konstruktivistického přístupu k vzdělávání. Moodle je poskytován zdarma jako Open Source software (spadající pod [obecnou veřejnou licenci GNU](#)). To v zásadě znamená, že je chráněn autorskými právy, ale poskytuje přitom uživatelům značnou svobodu. Moodle můžete kopírovat, používat i upravovat, pokud souhlasíte s tím, že: budete tento zdroj poskytovat ostatním; nebudete měnit ani odstraňovat původní údaje o licencích a autorských právech, a uplatníte stejné licenční podmínky i u jakýchkoliv odvozených produktů. Moodle lze použít na jakémkoliv počítači s fungujícím PHP, podporuje řadu typů databází (především MySQL). Důležitou součástí projektu Moodle je webová stránka [moodle.org](#). Je zdrojem informací a místem pro diskusi a spolupráci uživatelů, mezi něž patří správci systémů, pedagogové, metodici, vědci, a samozřejmě vývojáři.

Obr. 17 Úvodní stránka Moodle

Databáze Moodle je na serveru <http://vyuka.ft.utb.cz>. Na tomto serveru je několik kurzů. Administrátor přiděluje práva k jednotlivým kurzům. Než tedy začneme upravovat některý kurz musíme k němu mít povolen přístup. Cílem mé bakalářské práce je vytvořit test pro předmět Mikropočítače, proto musím mít právo tento kurz modifikovat.

2.2.2 Úvod k tvorbě testů

Obr. 18 Úvodní stránka kurzu Mikropočítače

Při přístupu na server je nutné se přihlásit, výběrem položky *Přihlásit se*, v horním pravém rohu obrazovky. Zadáním uživatelského jména, pod kterým jsou jednotliví uživatelé zaregistrováni v systému, a platného hesla. Pak si z nabídnutých kurzů zvolíme požadovaný kurz. Kurz Mikropočítače je ve skupině kurzů bakalářského studia. V nabídce činností, v levé části, volbou možnosti *Testy*, se zobrazí výpis všech již vytvořených testů tohoto kurzu.

Kategorie	Podrobnosti o kategorii	Zveřejnit	Úlohy	Akce
Standard	Přednastavená kategorie úloh.	Ne	0	
Dolinay1		Ano	38	Odstranit
zkouška		Ano	18	Odstranit
zkouška1		Ano	17	Odstranit
		Ne		Přidat

Uložit změny

Obr. 19 Úprava kategorie

Výběrem možnosti *Upravit úlohy*, nám je nabídnut výběr kategorie. Po vytvoření kategorie, si ji vybereme v pop-up menu. V opačném případě volbou *Upravit kategorii*, jsou vypsány všechny existující kategorie. Poslední řádek je volný, na toto místo se zadává název kategorie, která má být vytvořena. Určuje se také zda má být kategorie zveřejněna. Pro vytvoření kategorie potvrzujeme volbu *Uložit změny*.

Po úspěšném vytvoření kategorie, potvrzené vypsání textu „Kategorie (název kategorie) byla přidána“, volbou *Zpět k úpravě testu*, se zobrazí předchozí stránka. Volbou kategorie v pop-up menu, je zobrazena podobná stránka, jen místo názvu úlohy je text „Zatím žádné úlohy nebyly vloženy do textu“.

Mikropočítače Odhlásit se

Výuka na IIT » MP » Testy » Upravit úlohy

Kategorie: zkouška (MP) Upravit kategorii

Vytvořit novou úlohu: Vybrat... ?

Importovat úlohy ze souboru ?

Exportovat úlohy do souboru ?

Vytvořit více úloh najednou ?

Název úlohy	Typ	Upravit
Zásobníková paměť je typu		
Jakou zkratkou se označuje příznakový registr ?		
Jsou dva druhy přerušení?		
Má pes dvě nohy?		
Je HC11 jednočipový mikropočítač?		

Obr. 20 Úprava úlohy

Úlohy jsou do testu vkládány potvrzením možnosti Vytvořit novou úlohu, v pop-up menu jsou na výběr tyto možnosti - Úloha s výběrem odpovědi, Pravda/nepravda, Krátká odpověď, Numerická úloha, Vypočteno, Přiřazování, Doplnovací úloha, Přiřazování pro náhodně vybrané úlohy se stručnou odpovědí, Náhodně vybraná úloha, Popis.

2.2.3 Typy úloh

Tabulka II Přehled ikon jednotlivých typů úloh

	Úloha s výběrem odpovědi
	Krátká odpověď
	Pravda/nepravda
	Numerická
	Přiřazování
	Přiřazování pro náhodně vybrané úlohy
	Náhodně vybraná úloha

Úloha s výběrem odpovědi

Při odpovědi na tuto otázku student vybírá odpověď z několika možností. Tato otázka může mít jednu nebo více možných odpovědí. Pokud má jen jednu správnou odpověď je vhodné ji ohodnotit 100%, popř. jí můžeme přiřadit jiné procentuální ohodnocení. V případě že více odpovědí je správných může být každá odpověď obodována pozitivním i negativním procentuálním ziskem bodů. Vybere-li student všechny možnosti, nemusí nutně získat dobré hodnocení. Je-li součet bodů za tuto úlohu záporný, jako výsledná hodnota se dosadí nula.

Krátká odpověď

Jako odpověď na tuto otázku student napíše jedno slovo nebo sousloví. Na otázku může existovat více správných odpovědí, a to s různým bodovým hodnocením. U odpovědi lze nastavit, zda se budou rozlišovat malá a velká písmena. U odpovědi hraje velkou roli přesnost, takže dejte pozor na pravopis! I tato úloha může obsahovat obrázek.

Numerická úloha

Z pohledu studenta vypadá numerická úloha stejně jako otázka s krátkou odpovědí. Rozdíl je v tom, že u numerické úlohy je stanovena přijatelná chyba. Může být definován souvislý interval odpovědí, které jsou považovány za správné. Jestliže je například správná odpověď 30 s přijatelnou chybou 5, pak budou všechna čísla mezi 25 a 35 akceptována jako správná odpověď. Numerické úlohy mohou mít i nečíselné odpovědi, u těch se pak nerozlišuje velikost písmen. To může být užitečné, je-li odpovědi na numerickou úlohu slovo, slovní spojení, zkratka apod.

Pravda / nepravda

U této úlohy je studentovi předloženo tvrzení a jeho úkolem je rozhodnout, zda je pravdivé či nepravdivé. Pokud je nastaveno zobrazování komentářů, studentovi se po odeslání testu zobrazí odpovídající komentář. Pokud je například správná odpověď "nepravda" a student odpověděl "pravda" (otázku tedy zodpověděl nesprávně), pak se mu zobrazí komentář k odpovědi "pravda". Také tato úloha může obsahovat obrázek.

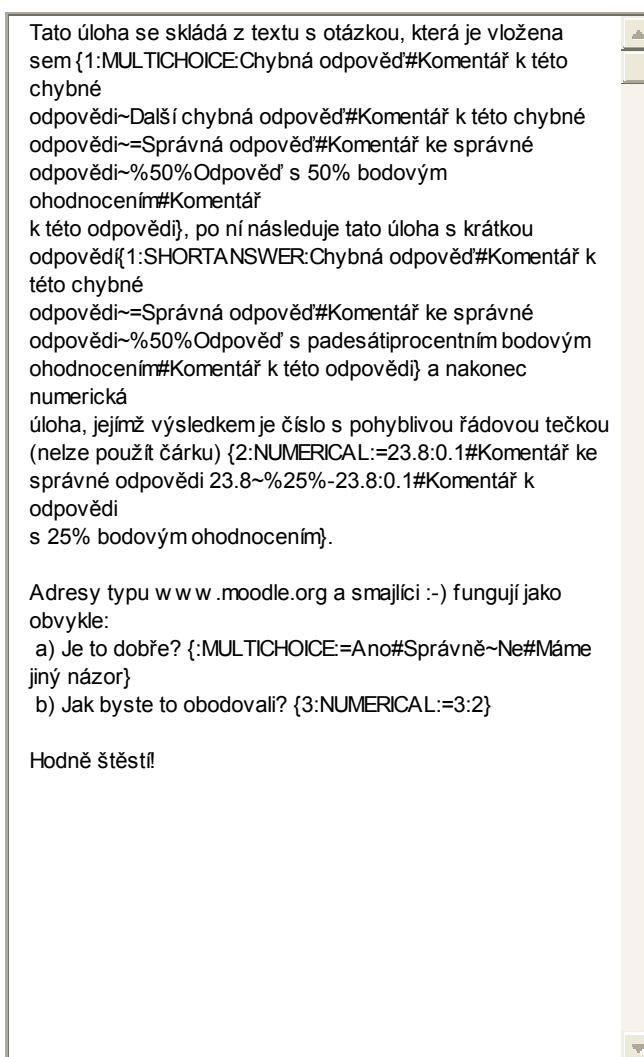
Přiřazování

Úloha může obsahovat krátký úvodní text, poté jsou studentovi předloženy otázky a seznam možných odpovědí. Ke každé otázce musí student přiřadit správnou odpověď, existuje vždy jen jedna. Všechny otázky mají stejnou váhu.

Doplňující úloha

Každá úloha sestává z textu, do kterého jsou zabudovány otázky různého typu (úloha s výběrem odpovědi, krátká odpověď, numerická úloha). Pro tvorbu těchto úloh momentálně není k dispozici grafické rozhraní. Úlohy je nutné tedy zadávat jako text s vloženými řídicími příkazy, a to buď pomocí vestavěného editoru, nebo z předem připravených souborů.

Příklad zadání úlohy:



Tato úloha se skládá z textu s otázkou, která je vložena sem {1:MULTICHOICE:Chybná odpověď#Komentář k této chybné odpovědi~Další chybná odpověď#Komentář k této chybné odpovědi~Správná odpověď#Komentář ke správné odpovědi~%50%Odpověď s 50% bodovým ohodnocením#Komentář k této odpovědi}, po ní následuje tato úloha s krátkou odpovědí {1:SHORTANSWER:Chybná odpověď#Komentář k této chybné odpovědi~Správná odpověď#Komentář ke správné odpovědi~%50%Odpověď s padesátiprocentním bodovým ohodnocením#Komentář k této odpovědi} a nakonec numerická úloha, jejímž výsledkem je číslo s pohyblivou řádovou tečkou (nelze použít čárku) {2:NUMERICAL:=23.8:0.1#Komentář ke správné odpovědi 23.8~%25%-23.8:0.1#Komentář k odpovědi s 25% bodovým ohodnocením}.

Adresy typu www.moodle.org a smajlíci :-) fungují jako obvykle:

a) Je to dobře? {1:MULTICHOICE:=Ano#Správně~Ne#Máme jiný názor}

b) Jak byste to obodovali? {3:NUMERICAL:=3:2}

Hodně štěstí!

Tento příklad se zobrazí studentům následovně:

3	Tato úloha se skládá z textu s otázkou, která je vložena sem
8 bodů	, po ní následuje tato úloha s krátkou odpovědí a nakonec numerická úloha, jejímž výsledkem je číslo s pohyblivou řádovou tečkou (nelze použít čárku) .
	Adresy typu www.moodle.org a smajlíci 😊 fungují jako obvykle:
	a) Je to dobře?
	b) Jak byste to obodovali?

Přiřazování pro náhodně vybrané úlohy se stručnou odpovědí

Z pohledu studenta vypadá tato úloha jako *Přiřazování*. Rozdíl je v tom, že jednotlivé otázky jsou náhodně vybrány z fondu otázek se stručnou odpovědí v aktuální kategorii. Úloha může obsahovat krátký úvodní text, poté je studentovi předloženo několik otázek a seznam možných odpovědí. Pro každou otázku existuje právě jedna správná odpověď. Student musí přiřadit odpověď ke každé otázce. Při výpočtu celkového skóre za úlohu mají všechny otázky stejnou váhu. Otázky jsou náhodně vybrány z fondu otázek s krátkou odpovědí v aktuální kategorii. Studentovi je při každém pokusu o řešení testu předložena jiná sada otázek.

Náhodně vybraná úloha

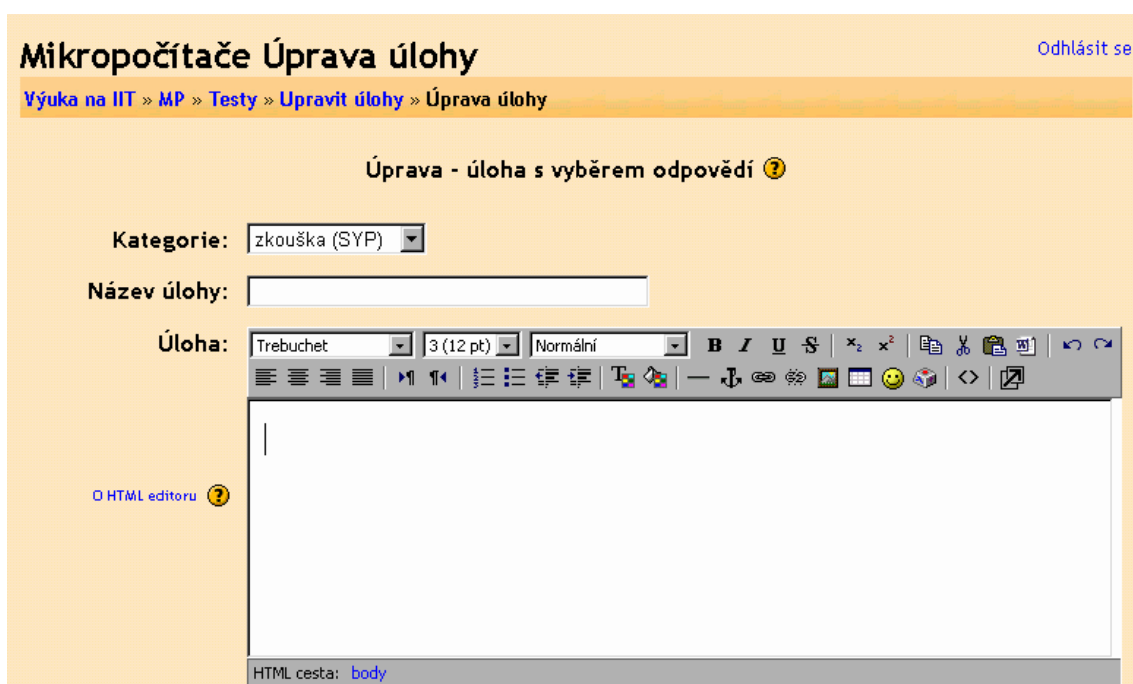
Náhodně vybraná úloha nepředstavuje vlastně samostatný typ úlohy. Pokud tuto úlohu v testu použijete, bude na její místo při každém studentově pokusu dosazena náhodně vybraná úloha ze zvolené kategorie. Maximální počet bodů za tuto úlohu bude takový, jaký jste stanovili při vkládání náhodné úlohy do testu (tj. nebude se brát ohled na bodové ohodnocení původně stanovené pro dosazovanou otázku). Jestliže například sestavíte test s deseti náhodnými úlohami, každý student může při každém pokusu dostat

k řešení úplně jinou sadu deseti otázek. Náhodně vybrané úlohy lze kombinovat s nenáhodnými. Tím můžete zajistit, že určité úlohy budou do testu zařazeny pokaždé.

Popis

Popis není skutečná úloha. Slouží k zobrazení libovolného textu (případně obrázku) a nevyžaduje odpověď. Lze jej použít například k zobrazení informací společných pro následující skupinu otázek.

2.2.4 Nastavení úlohy



Obr. 21 Úprava úlohy – zadání textu otázky

Po výběru typu úlohy se zobrazí stránka, kde jsou úlohy editovány. Zadává se název úlohy. Pokud není vyplněn, je automaticky jako název označen text otázky. HTML editor poskytuje rozhraní pro zpracování textu. Toto rozhraní je vloženo do webové stránky a umožňuje upravovat text jako v jiných editorech, přičemž produkuje normální HTML kód. Lze vkládat text z jiných aplikací systému Windows. Text se zkopíruje do schránky a pak může být vložen do tohoto editoru s tím, že formátování textu zůstává zachováno. Je možné také vkládat obrázky a to z jakékoliv webové stránky stisknutím tlačítka vložit obrázek ikonou na pracovní liště editoru. Vedlejší ikona plní funkci vkládání tabulky. K vytvoření odkazu nejprve napište text, který má být použit jako odkaz. Označíme jej a kliknutím na příslušné tlačítko v nástrojové liště, která vypadá jako část

řetězu a je zadávána odkazovanou adresu URL. Můžeme také vkládat tzv. smajlíky (emotikony), stisknutím ikony smajlíka.

Obr. 22 Úprava úlohy – zadávání odpovědí

Při volbě *Úlohy s výběrem odpovědí* je nutné zadat kolik odpovědí je správných - pouze jedna odpověď nebo více odpovědí. Následovně při zadávání možností odpovědí, musí být vyplněny alespoň dvě, jinak nebude otázka vytvořena. Vedle odpovědí jsou zadávány body.

Je-li pouze jedna odpověď správná, měla by být ohodnocena 100%, není to však podmínkou. Pokud jsme stanovili více správných odpovědí, můžeme je obodovat tak, aby jejich součet dával 100%, nebo jedna správná odpověď může mít 100% a zbylé libovolný počet procent.

Vytvořené otázky se zobrazují pod svým názvem. Vedle názvu je také zobrazen typ této úlohy a možnosti oprav. Přehled ikon odpovídajících jednotlivým typům úloh je v tabulce I. Pro odstranění úlohy, potvrdíme ikonu křížku, která je v úpravách zobrazena jako první možnost. Systém se nás ještě zeptá, zda úlohu opravdu chceme odstranit. Další možností úprav je náhled otázky. Zobrazí se nám, jak bude otázka vypadat v testu. Také si můžeme ověřit, jestli jsme zadali správně odpověď tím, že označíme nějakou odpověď a

potvrdíme *Zkontrolovat*. Vypíše se nám odpovědi, zeleně je označena správná odpověď a u odpovědi, která byla označena jako správná je pomlčka. Poslední možnou úpravou je editace. Kde se nám objeví stejná stránka, jako při vytváření této úlohy. Jsme-li spokojeni s úpravami, potvrdíme volbu *Uložit* ve spodní části obrazovky.

Importovat úlohy ze souboru

Tato funkce umožňuje importování úloh z externích textových souborů. Moodle však nepodporuje všechny formáty. Můžeme vkládat pouze soubory formátu GIFT, vynechané slovo, Suken, Aon, Blacboard, Course Test Manager popřípadě uživatelského formátu.

GIFT je nejkomplexnější dostupný formát pro importování testových otázek z textového souboru do Moodlu. Byl navržen pro snadný textový zápis testových otázek. Podporuje různé typy úloh: úlohy s výběrem odpovědí, s krátkou odpovědí, s odpovědí pravda/nepravda, numerické úlohy a přiřazování. V jednom textovém souboru lze použít více typů úloh. Formát GIFT dále umožňuje zadávat jméno otázky, komentář k odpovědím a jejich hodnocení v procentech; pro interní potřebu lze text také opatřovat poznámkami, které se v testu nezobrazí.

Zde je několik příkladů:

Kdo je pohřben v památníku na Vítkově? {~T.G.M. ~Klement Gottwald =nikdo}

V památníku na Vítkově byl {~otráven =pohřben ~oddán} Klement Gottwald.

V památníku na Vítkově je pohřben Klement Gottwald. {FALSE}

Kdo je pohřben v památníku na Vítkově? {=nyní již nikdo =nikdo}

Kdy se narodil Klement Gottwald? {#1896}

Formát Aiken umožňuje snadnou tvorbu úloh s výběrem odpovědí. Používá jednoduchou syntaxi. Otázka musí být napsána celá na jednom řádku. Každá varianta odpovědi musí být uvozena jedním písmenem (A, B, C...), po němž následuje tečka "." nebo závorka ")", a za nimi mezera. Pak musí bezprostředně následovat řádek se správnou odpovědí.

Příklad :

Co je cílem první pomoci?

A. Zachránit život, zabránit dalšímu zranění

B. Poskytnout lékařské ošetření zraněnému

C. Zabránit dalšímu zranění

D. Pomoci obětem v nouzi

ODPOVĚĎ: A

Který LMS nabízí nejvíce importních. formátů pro testy?

A) Moodle

B) Ilias

C) Claroline

D) Blackboard

ANSWER: A

Formát Vynechané slovo podporuje pouze úlohy s výběrem odpovědí. Chybné varianty odpovědi se uvozují vlnovkou (~), správná odpověď rovnítkem (=).

Příklad :

Již v raném věku, kdy začínáme zkoumat své tělo, se z nás stávají studenti

{=anatomie a fyziologie ~mechaniky ~teologie ~botaniky}, a studenty v jistém smyslu zůstáváme až do konce života.

Formát AON se od předchozího liší pouze tím, že po provedení importu se všechny úlohy s krátkou odpovědí po čtveřicích převedou na přiřazovací úlohy. Kromě toho se u úloh s výběrem odpovědí během vkládání náhodně změní pořadí odpovědí.

Importovat lze také úlohy z exportního formátu systému Blackboard. Tento formát využívá funkcí PHP určených pro práci s jazykem XML.

Další možností importování úlohy z databáze testovacích otázek systému Course Test Manager (CTM). Pro přístup k testovacím otázkám (uloženým v databázi Microsoft Access) se používají různé postupy v závislosti na tom, zda je Moodle provozován v operačním systému Windows nebo Linux. Ve Windows můžete importovat databázi formátu Microsoft Access běžným způsobem. V případě Linuxu potřebujete počítač s Windows umístěný na stejné síti. Na něj zkopírujte databázi CTM a spusťte ODBC Socket

Server, který pro převod dat do Moodle na serveru Linux použije jazyka XML. Před použitím tohoto importního filtru si prosím přečtete podrobnější nápovědu.

Pokud máte vlastní formát, který potřebujete importovat, upravte podle potřeby soubor `mod/quiz/format/custom.php`. Rozsah nutných změn v kódu je opravdu malý. Nový kód je potřeba pouze pro rozpoznání jedné úlohy ve výchozím textu.

Zvolíme-li importování úloh ze souboru, systém po nás chce zadat formát souboru, zvolíme si v pop-up menu. Volbou *procházet* zvolíme daný soubor a potvrdíme, že chceme vložit soubor.

Exportovat úlohy do souboru

Doporučuje se exportovat pouze úlohy formátu GIFT, který byl popsán v předchozím odstavci. Systém po nás vyžaduje zadat kategorii, formát souboru. Jméno souboru je vypsané automaticky, můžeme jej změnit.

Vytvořit více úloh najednou

Vytvořit více úloh najednou ?

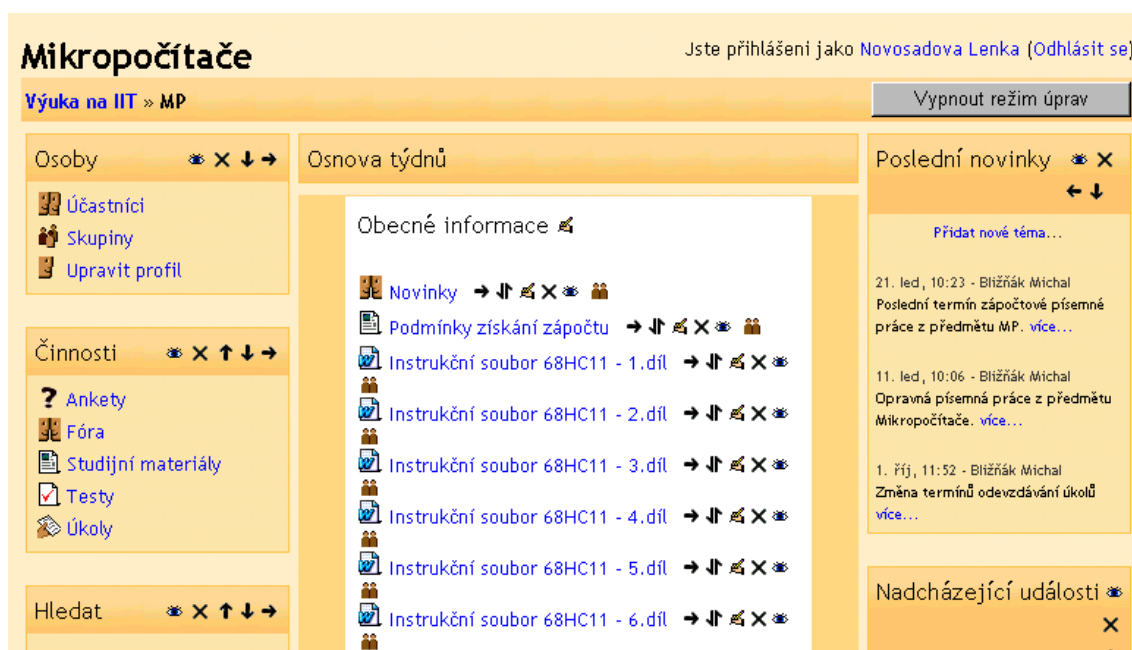
Kategorie:	zkouška (SYP) ▼
Vytvořit náhodně vybrané úlohy:	10 ▼
Standardní počet bodů za úlohu:	1 ▼
Přidat úlohy do stávajícího testu:	Ano ▼
Vytvořit více úloh najednou	

Obr. 23 Úprava úloh – vytvořit více úloh najednou

Zadáваме kategorii testu do kterého chceme otázky vytvořit. Vytvoří se náhodně vybrané úlohy, jejich počet si můžeme zvolit v rozmezí 1 – 100. Také se nás zeptá, zda chceme vytvořené úlohy vložit do stávajícího testu a kolik bodů se má daným úlohám standartně přiřadit. Potvrdíme možnost *Vytvořit více úloh najednou*. V kategorii tedy budou vytvořené vstupní otázky a také příslušný počet otázek typu *Náhodně vybraná úloha*, jejich počet však nemůže být větší než počet vstupních otázek.

2.2.5 Zavedení testu do systému

Máme-li test vytvořen, dáme možnost Pokračovat. Tím se dostaneme do správy testů. Chceme-li svůj test přidat do systému, musíme se vrátit na úvodní stránku kurzu. V našem případě Bakalářské studium / Mikropočítače. Na této stránce *Zapneme režim úprav*. Tato možnost se nám nabízí v pravém, horním rohu obrazovky. Následně se nám u jednotlivých činností, které již jsou do kurzu vloženy, jako jsou jiné testy, úkoly, studijní materiály atd., zobrazí možnosti úprav. Stránka může vypadat následovně viz. obr..



Obr. 24 Zapnut režim úprav

Musíme se rozhodnout, do kterého týdne chceme daný test zařadit. Vzhledem k tomu, že vytvářím test, ve kterém se testují znalosti studentů za celý semestr, vložila jsem tento test do posledního týdne. V tomto týdnu si vyberu v pop-up menu nazvaném přidat činnost *Test*. Zobrazí se stránka, kam se zadává jméno testu např. závěrečný test. Do části označené *Úvod*, můžeme vepsat libovolný text. V našem případě je zde popis testu a jeho hodnocení. Tento text se zobrazí při vykonávání testu.. Práce s editorem byla popsána v části *Nastavení úlohy*.

The screenshot shows a configuration form for a test. The settings are as follows:

- Zpřístupnit test:** 19 září 2004 23:00
- Uzavřít test:** 27 září 2004 00:00
- Časový limit:** Žádný
- Zamíchat úlohy:** Ne
- Zamíchat odpovědi:** Ano
- Dovoleno pokusů:** Neomezený počet pokusů
- Každý pokus staví na předchozím:** Ne
- Metoda bodování:** Nejvyšší počet bodů (dropdown menu is open showing options: Nejvyšší počet bodů, Průměrný počet bodů, První pokus, Poslední pokus)
- Ukázat:** (partially visible)

Obr. 25 Přidání činnosti I

U testu se zadává časové rozmezí, kdy je test zpřístupněn studentům. Časové rozmezí se nastavuje v kolonkách *Zpřístupnit a Uzavřít test*. Nastavuje se zde den, měsíc, rok (prozatím lze nastavit do roku 2010) a také hodiny a minuty. Můžeme tedy zpřístupnit test třeba jen na pět minut. Musíme však dbát na to, aby studenti stačili v tomto čase test napsat. Můžeme také zadat časový limit, který student má na vykonání testu. Lze zadat v rozmezí jedné a stodesetin minut. Při vykonávání testu je zobrazeno, kolik času zbývá do skončení testu. Jakmile zbývá jen 15 sekund, začne pozadí u zobrazovaného času postupně červenat. Po vypršení časového limitu je test uzavřen.

Nastaví se zda mají být otázky či odpovědi zamíchány. Pak pořadí úloh/odpovědí bude náhodně zamícháno. Studentům se také může povolit více pokusů na zvládnutí testu, jeden až neomezeně pokusů. V našem testu byla vybrána možnost tří pokusů, vzhledem k tomu, že student má nárok na tři pokusy zvládnutí zkoušky. Zamezíme tak studentům neomezený počet pokusů a máme větší přehled kolikrát se již zkoušky účastnili.

Také můžeme zadat, aby se *Každý pokus stavil na předchozím*. Pokud studentovi povolíte při testu více pokusů a u tohoto nastavení zvolíte ANO, bude test při každém novém pokusu obsahovat všechny odpovědi i zpětnou vazbu z pokusů předchozích. Chceme-li, aby se studentovi při každém pokusu ukázal prázdný test, zvolte u tohoto nastavení NE.

Nastavení metody bodování má smysl v případě, že student má více pokusů na vykonání testu. Volbou *Nejvyšší počet bodů*, kdy v konečném hodnocení je nejvyšší počet bodů, který student získal ze všech pokusů. Nebo *Průměrný počet bodů*, kdy se zobrazí průměrný počet bodů vypočtených ze všech pokusů. Nebo se zobrazují jen body dosažené z první pokusu. Pak jsou další pokusy zbytečné. A nakonec možnost *Poslední pokus*, kdy je jako konečné hodnocení použit počet bodů, kterého student dosáhl při posledním pokusu. Tuto variantu jsem zvolila také – student je buď spokojen s výsledkem, nebo píše test znovu, bez ohledu na to jakého výsledku dosáhl.

The screenshot shows a configuration window with an orange background. It contains several settings, each with a label, a dropdown menu, and a help icon (a yellow circle with a question mark). The settings are:

- Ukázat hodnocení po odpovědi?:** The dropdown menu is set to "Ne".
- Ukázat správnou odpověď v hodnocení?:** The dropdown menu is set to "Ne".
- Umožnit prohlédnutí:** The dropdown menu is set to "Ne".
- Maximální počet bodů:** The dropdown menu is set to "Žádné body".
- Vyžaduje heslo:** An empty text input field.
- Vyžaduje síťovou adresu:** An empty text input field.

At the bottom right of the configuration area is a button labeled "Pokračovat".

Obr. 26 Přidání činnosti II

Při nastavení také můžeme povolit zobrazení komentáře neboli vysvětlení, volbou ANO u možnosti *Ukázat hodnocení po odpovědi*. Studentům se ke každé odpovědi zobrazí komentář (ať byla odpověď správná, či chybná). Dále volíme zobrazování správné odpovědi v hodnocení. Umožníme-li zobrazení správných odpovědí, bude studentům u každé otázky zobrazena správná odpověď (zvýrazněná zelenou barvou). Umožníme-li prohlédnutí, studenti si budou moci prohlédnout všechny své předchozí pokusy o řešení, ale až po uzavření testu.

Dále se zadává maximální počet bodů, který student může v testu získat, v rozmezí 0 – 100 bodů. U volby *vyžaduje heslo* smíme zadat heslo, pod kterým se test zpřístupní. Jsme-li s nastavením testu spokojeni potvrdíme *Pokračovat*.

Výuka na IIT » MP » Testy » Úprava testu

ZÁVĚREČNÝ TEST

Pořadí	Název úlohy	Typ	Body	Upravit
1	Náhodně vybraná úloha	?	2	x

Uložit obodování: 1

Uložit celý test Zrušit

Kategorie: zkusební (MP) Upravit kategorii

Vytvořit novou úlohu: Vybrat...

Importovat úlohy ze souboru
Exportovat úlohy do souboru
Vytvořit více úloh najednou

Vybrat	Název úlohy	Typ	Up
☐	Kdo je garantem předmětu Mikropočítače?	≡	x
☐	Náhodně vybraná úloha	?	x

<< Přidat vybranou do testu Vybrat vše

Obr. 27 Úprava testu

Zobrazí se stránka rozdělená na dvě poloviny. V levé části vidíme test, který vytváříme nebo upravujeme, v horní části je jméno tohoto testu. V pravé části se volí kategorie, ze které chceme vkládat úlohy. Většinou chceme vkládat úlohy, které jsme si vytvořili. Můžeme si však vybrat i kategorii někoho jiného, pokud tvůrce kategorií zveřejnil. Je také možné použít úlohy z více kategorií. A teď už to je úplně snadné. Ve výpise kategorie se označí vybrané úlohy, které mají být zařazeny do testu. Pod touto nabídkou nám jsou nabídnuty dvě možnosti – *Vybrat vše* a *Přidat vybranou do testu*.

Při vkládání úloh, je důležité si pamatovat, že počet *Náhodně vybraných úloh* nesmí být větší než počet vytvořených vstupních úloh. Použijeme-li více náhodně vybraných úloh, měli by všechny mít stejné bodové ohodnocení, tyto úlohy se generují náhodně, nevíme, která úloha se přidělí dané náhodně vygenerované. Další možností je vytvořit více kategorií. V každé kategorii budou zadány vstupní úlohy za určitý počet bodů. Tedy v jedné kategorii budou otázky za jeden bod. Při vytváření *Více úloh najednou*, pak nastavíme standardní počet bodů za úlohu na 1. V další kategorii vytvoříme úlohy ohodnocené 2 body atd.. Jak již bylo řečeno, při tvorbě testu můžeme vkládat úlohy z více kategorií. Což nám umožňuje, aby náhodné úlohy měly různé bodové ohodnocení, podle toho jak jsme si je obodovali. Pokud chceme, aby některé otázky byly v testu vždy, je nejvhodnější vytvořit je v samostatné kategorii. Do testu je vkládáme rovnou bez použití možnosti *Náhodně vybrané úlohy*. Jinak by se v testu mohly objevit dvakrát. Musíme také dbát na to, že maximální počet získaných bodů nemůže přesáhnout 100 bodů.

Jakmile máme označeny úlohy, které mají být vloženy do testu, je akce provedena potvrzením možnosti *Přidat vybranou do testu*. Stačí jednotlivým otázkám přiřadit body a potvrdit volbu *Uložit celý test* a tento test je uložen do systému.

Test se může přenastavit na úvodní stránce kurzu, potvrzením volby *Zapnout režim úprav*. Potvrzením ikony editace u jména testu. Zobrazí se stránka Úprava testu, která byla popsána výše, v části *Zavedení testu do systému*.

Tabulka III Přehled ikon v Moodle

Ikona	Označení ikony	Stručný popis
	nápověda	otevře nové okno s nápovědou
	otevřené oko	skryje položku před studenty
	zavřené oko	označuje skrytou položku a umožňuje ji zpřístupnit
	editace	úprava položky
	úkoly	student sem odevzdává svou práci
	testy	označení testu
	odstranit	vymazání položky
	přesunout	uložení na jinou pozici
	náhled	náhled

2.3 Srovnání testovacích programů

2.3.1 Výhody a nevýhody aplikace Testy

Tato aplikace splnila námi zadané požadavky pouze z menší části. Hlavní, co bych aplikaci vytkla, je nedostatečná nápověda (popis aplikace). Pro začínajícího uživatele je trochu náročné se zorientovat. Ke kladným vlastnostem této aplikace bych přiřadila výpis IP adresy počítače, na kterém je test vykonán. Na druhou stranu, komu by se chtělo pokaždé kontrolovat, zda to odpovídá. Ve výsledcích je uvedeno, jak student odpovídal na jednotlivé otázky. Výhodou je u výpisu výsledků, že mohou být seříděny a to podle zkoušeného, jména testu, ročníku, studijní skupiny, bodů či podle klasifikace.

Za hlavní výhodu považuji to, že se při vyhodnocení testu určila také klasifikace. Tato funkce mi však téměř vůbec nefungovala. Vytvářela jsem testy s postupně navyšujícím počtem otázek. U všech testů, které jsem vytvořila, mi fungovalo pouze hodnocení *výborně* a *nevyhověl*. Navíc je v testu klasifikace rozdělena jen do tří skupin. Nyní je klasifikační stupnice rozdělena již do pěti skupin. Při psaní testu pak student zadává sám své jméno a skupinu. To nepovažuji jako nejlepší řešení, protože zkoušející nemá úplný přehled o všech svých studentech. Mohl by tedy test vykonat někdo jiný. Nevhodně je řešeno vymazání testů. Výsledky zůstanou v databázi do té doby, než je odtud vymažeme. Záznamy se dají vymazat pouze po jednom, což může být časově náročné při větším počtu studentů.

Při vytváření testů jsme značně omezováni. Již při nastavení parametrů testu, kdy test může obsahovat maximálně 40 otázek. Dále nemůžeme změnit přístupové heslo testu. Pokud se student heslo doví, např. při prvním pokusu o zvládnutí testu, má k tomuto testu přístup bez omezení. Může si jej spustit kdekoliv a my o tom vůbec nemusíme vědět, nebo při odevzdání testu může zadat jméno někoho jiného. Test studenta nebude uložen do databáze, pokud jeho jméno ve výpise daného testu už je. Při nastavení testu jsou napevno zadány jeho parametry. Jediné co můžeme po vytvoření testu změnit je klasifikace.

2.3.2 Výhody a nevýhody vytváření testů v systému Moodle

Moodle je rozšířený výukový systém, který využívá velká skupina uživatelů a neustále se vyvíjí. Při vyskytnutí nějakého problému se můžeme obrátit na webovou stránku moodle.org. Tvorba testů je relativně snadná, nabízí se velká škála možností. Při

vytváření úloh je na výběr několik typů. Úloha může mít více správných odpovědí, které mohou být ohodnoceny částečným ziskem bodů. Za špatnou odpověď se část bodů může odečíst. Při tvorbě testu, jsou vytvářeny kategorie, v každé může být až sto úloh. V počtu kategorií nejsme omezováni. Můžeme nastavit, aby test obsahoval náhodně vybrané úlohy, které mohou být náhodně generovány z více kategorií.

U nastavení testu lze omezit přístup. Zadáním datumu a čas, kdy je test dostupný studentům. Test může být zpřístupněn třeba jen na několik minut. Nebo omezit přístup k testu přístupovým heslem. Vykonání testu může být také omezeno časovým limitem. Po jeho uplynutí jsou výsledky testu uloženy a test ukončen. Test může být přenastaven kdykoli, s tím že se dají změnit všechny jeho parametry. Studentům přiřazujeme určitý počet pokusů o zvládnutí testu. Všechny jeho pokusy jsou zaznamenány. Všichni studenti jsou v systému zaregistrováni. Mají přístup pod svým uživatelským jménem – příjmení studenta a heslem. Ve výsledcích je tedy uvedeno jejich jméno a dosažený počet bodů.

2.3.3 Výběr aplikace

Výše byly shrnuty výhody a nevýhody aplikací. Na základě požadavků na test, které byly stanoveny před vytvořením testu, byl test vytvořen v databázi Moodle. Pro tuto možnost jsem se rozhodla proto, že otázky jsou náhodně generovány. Výhodou je také to, že parametry testu včetně hesla se dají upravovat kdykoliv. Test může být zpřístupněn jen na omezenou dobu. Je také lépe zabezpečeno, aby se k testu dostali jen ti studenti, pro které je určen. Každý student, který má přístup k databázi, má vytvořen účet pod svým jménem. Jsou tedy zaznamenána jména všech studentů, kteří se o přístup k testu pokusí.

Test je přístupný po zadání hesla. Obsahuje třicet otázek. Časový limit na zvládnutí testu je nastaven na 20 minut. Každý student má na zvládnutí testu tři pokusy. Otázky jsou generovány náhodně a jsou typu Úloha s výběrem odpovědi a Krátká odpověď. Maximální počet získaných bodů je 100.

ZÁVĚR

Zadáním bakalářské práce je vytvořit „**Elektronickou pomůcku pro předmět Mikropočítače**“ .

První část bakalářské práce je zaměřena na problematiku programového vybavení mikropočítačů. Zde jsou popsány jejich jednotlivé součásti včetně popisu jejich funkce. Práce je zaměřena na mikropočítače HC11 firmy Motorola.

Vybrané kapitoly jsou zpracovány jako prezentace v PowerPointu. Jsou součástí bakalářské práce v elektronické podobě viz. příloha: P II – CD-ROM.

Druhá část bakalářské práce je zaměřena na prostudování programového vybavení pro testování znalostí studentů. Byly porovnávány dvě aplikace. První aplikace se nachází na školním serveru <http://testy.utb.cz>. Tato aplikace je vytvořena ve skriptovacím jazyce PHP a pracuje s databází MySQL. Druhá aplikace je součástí databáze Moodle, která je na serveru <http://vyuka.ft.utb.cz>. Na základě porovnání těchto aplikací byl vytvořen test v databázi Moodle, která jednoznačně splnila nejlépe stanovené požadavky.

Vytvořený test je rovněž součástí bakalářské práce a to v tištěné podobě viz. příloha: P I – Vytvořený test. Přístup k testu mají všichni studenti, kteří jsou zapsáni v kurzu Mikropočítačů v databázi Moodle, po zadání hesla testu.

SEZNAM POUŽITÉ LITERATURY

- [1] VAŠEK, V. Pomůcky pro přednášky z předmětu Mikropočítače. Interní pomůcky FT UTB, Zlín 2005.
- [2] ROZEHNAL, Z. Mikrokontroléry HC 11. BEN - technická literatura, Praha 2001.
- [3] VAŠEK, V., VAŠEK, L. Programování mikropočítačů. FT VUT v Brně, Zlín 1989.
- [4] VLACH, J. Počítačová rozhraní. BEN technická literatura, Praha 2000.
- [5] VLACH, J. Řízení a vizualizace technologických procesů. BEN technická literatura, Praha 1999.
- [6] PECHAL, S. Monolitické mikropočítače. BEN technická literatura, Praha 1998.
- [7] SROVNAL, V. Operační systémy pro řízení v reálném čase, VŠB, Ostrava 2003.
- [8] VESELÝ, J. Testy a testové otázky na WEB stránkách. Diplomová práce FT UTB, Zlín 2002.
- [9] VAŠEK, V. Mikropočítačové monitorovací a řídicí systémy. FT VUT v Brně, Zlín 1994.
- [10] ZDENĚK, J. Monolitické mikropočítače řady 51. MBE, Praha 1990.
- [11] MINICH, P. Programové vybavení pro mikropočítačový řídicí systém. Diplomová práce FT VUT, Zlín 1991.
- [12] PINKER, J. Mikroprocesory a mikropočítače. BEN technická literatura, Praha 2004.
- [13] www.spszl.cz/svs/export.php?fid=78 [cit. 2005-03-05].
- [14] <http://lucy.troja.mff.cuni.cz/~tichy/kap9/mikropo.html> [cit. 2005-03-20].
- [15] <http://lucy.troja.mff.cuni.cz/~tichy/kap9/jednocpoc.html> [cit. 2005-03-20].
- [16] <http://www.svethardware.cz/glos.jsp>
- [17] <http://x-kure.wz.cz/Kniha/08.html>
- [18] http://www.kemt.fei.tuke.sk/predmety/%20KEMT411_ESM/_materialy/essm.pdf [cit. 2005-01-13].

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

CPU	- centrální procesorová jednotka
SPI	- sériově synchronní kanál, přerušení od periferního sériového kanálu
SCI	- sériově asynchronní kanál, přerušení od komunikačního sériového kanálu
IRQ	- maskovatelné přerušení
XIRQ	- nemaskovatelné přerušení
RTI	- přerušení od reálného času
HTML	- hypertextový značkovací jazyk pro tvorbu WWW stránek
CSS	- kaskádové styly k extrahování informace o stylech v HTML dokumentech
SQL	- strukturovaný dotazovací jazyk
PHP	- skriptovací jazyk zabudovaný na straně serveru
MySQL	- systém správy relačních databází využívající jazyka SQL
GIFT	- formát pro importování testových otázek z textového souboru do Moodle
LMS	- Learning Manager System
XML	- jazyk pro předávání informací mezi počítači se systémem Windows a počítači s jinými operačními systémy
CTM	- databáze testovacích otázek (<i>Course Test Manager</i>)

SEZNAM OBRÁZKŮ

Obr. 1 Blokové schéma HC11	11
Obr. 2 Struktura instrukčního slova	16
Obr. 3 Zdrojový řádek	16
Obr. 4 Průběh vykonávání procesů	19
Obr. 5 Úvodní stránka systému testy	24
Obr. 6 Psát test	25
Obr. 7 Psát test – odevzdání testu	26
Obr. 8 Správa testů - úvodní stránka	27
Obr. 9 Správa testů – nastavení uživatele	28
Obr. 10 Správa testů – výpis testů uživatele	28
Obr. 11 Správa testu – editace	30
Obr. 12 Správa testů – vkládání otázek	30
Obr. 13 Správa testů – založení testu	32
Obr. 14 Správa testů – výsledky testů	33
Obr. 15 Volné testy – psaní testu	35
Obr. 16 Volný test – vyhodnocení testu	36
Obr. 17 Úvodní stránka Moodle	37
Obr. 18 Úvodní stránka kurzu Mikropočítače	37
Obr. 19 Úprava kategorie	38
Obr. 20 Úprava úlohy	39
Obr. 21 Úprava úlohy – zadání textu otázky	43
Obr. 22 Úprava úlohy – zadávání odpovědí	44
Obr. 23 Úprava úloh – vytvořit více úloh najednou	47
Obr. 24 Zapnutí režim úprav	48
Obr. 25 Přidání činnosti I	49
Obr. 26 Přidání činnosti II	50
Obr. 27 Úprava testu	51

SEZNAM TABULEK

Tabulka I Přehled služeb systému pro řízení běhu procesu.....	21
Tabulka II Přehled ikon jednotlivých typů úloh	39
Tabulka III Přehled ikon v Moodle.....	52

SEZNAM PŘÍLOH

P I : Vytvořený test

P II : CD-ROM : Prezentace aplikace Microsoft PowerPoint - RS232C.ppt

- RTMON.ppt

- Převodníky.ppt

- OS RT.ppt

PŘÍLOHA P I: VYTVOŘENÝ TEST**Zbývajicí čas****0:24:52****závěrečný test****Pokus 2**

Závěrečný test z předmětu MIKROPOČÍTAČE

Může být více správných odpovědí, max. počet bodů je 90.

Časový limit - 25 minut.

hodnocení testu: 1 . . . 85 b.

-1 . . . 80 b.

2 . . . 70 b.

- 1** Číslo 111 vyjádřené ve dvojkové, osmičkové a hexadecimální soustavě je
3 Pokusů Odpověď: ☐ a. 1101111, 175, 6F
☐ b. 1101111, 157, 6F
☐ c. 110101, 157, 5C

- 2** Při použití makroinstrukce se PC registr naplní
3 Pokusů Odpověď: ☐ a. vždy stejnou vstupní adresou
☐ b. vypočtenou adresou na základě počtu volání makroinstrukce
☐ c. vždy jinou vstupní adresou

- 3** Přímo z běžícího programu můžeme změnit obsah paměťového místa v paměti typu
3 Pokusů Odpověď: ☐ a. EPROM
☐ b. EEPROM
☐ c. PROM

- 4** Jádro RTOS není spouštěno
3 Pokusů Odpověď: ☐ a. po službě create-proces
☐ b. po činnosti dispečera
☐ c. v okamžiku tiků RTOS

5 Při přechodu do podprogramu obsluhujícího přerušení s využitím vektoru přerušení a pseudovektoru přerušení se dostaneme

3 Pokusů

- Odpověď: ☐ a. přes jednu adresu
☐ b. přes dvě adresy
☐ c. přes tři adresy

6 Dvojkový doplněk čísla -12 je - zapsat bez mezer

3 Pokusů

Odpověď:

7 Vektor procesu je vytvořen

3 Pokusů

- Odpověď: ☐ a. automaticky systémem RTOS
☐ b. uživatel dává dílčí informace k jeho vytvoření
☐ c. je definován uživatelem

8 V rámci jednoho tiky neproběhne

3 Pokusů

- Odpověď: ☐ a. Režijní činnosti RTOS
☐ b. Je plánován proces, který se stane aktivním
☐ c. Definice vektoru procesů a systémových prostředků

9 U přerušovacího systému HC11 není implementováno

3 Pokusů

- Odpověď: ☐ a. maskovatelné přerušování Watchdog
☐ b. nemaskovatelné přerušování Watchdog
☐ c. maskovatelné softwarové přerušování

10 Na vzdálenost stovek metrů je možné přenášet data

3 Pokusů

- Odpověď: ☐ a. pomocí nulového modemu
☐ b. sériovým přenosem dat pomocí rozhraní RS232
☐ c. pomocí proudové smyčky

11 Povolení namaskovatelného přerušování mezi dvěma resety je možno provést

3 Pokusů

- Odpověď: ☐ a. vícekrát
☐ b. vícekrát, jedná-li se o vnořené přerušování
☐ c. jednou

12 Scheduler rozhoduje na základě

3 Pokusů

- Odpověď: ☐ a. tiků operačního systému
☐ b. vektorů přerušení
☐ c. priority čekajících procesů

13 Funkce Watch dog je realizována

3 Pokusů

- Odpověď: ☐ a. automaticky po spuštění jakéhokoliv programu
☐ b. hardwarově závislým čítačem periodicky nulovaným
☐ c. hardwarově nezávislým čítačem

14 Procesem rozumíme

3 Pokusů

- Odpověď: ☐ a. fungující programový modul
☐ b. programový modul s definovanou datovou strukturou
☐ c. činnost, která je provedena po sestavení

15 Instrukci vykonává

3 Pokusů

- Odpověď: ☐ a. překladač
☐ b. program
☐ c. procesor

16 V případě logické funkce exklusiv-or pro dvě proměnné A a B má výstupní proměnná C hodnotu 1, pokud

3 Pokusů

- Odpověď: ☐ a. A i B mají hodnotu 0
☐ b. A a B mají různou hodnotu
☐ c. A i B mají hodnotu 1

17 Služby RTOS jsou poskytovány

3 Pokusů

- Odpověď: ☐ a. pro využití možností RTOS pro aplikaci
☐ b. jádru OS pro sledování stavu jednotlivých procesů
☐ c. uživateli pro kontrolu správného běhu systému

18 Při překladu je zdrojový program zpracováván překladačem

3 Pokusů

- Odpověďt: ☐ a. jednou
☐ b. třikrát
☐ c. dvakrát

19 Součástí RTOS RTMONu není

3 Pokusů

- Odpověďt: ☐ a. dispatcher
☐ b. scheduler
☐ c. priority maker

20 Zdrojový řádek assembleru AS11 obsahuje pole?

3 Pokusů

- Odpověďt: ☐ a. návěští, instrukce, komentáře
☐ b. návěští, instrukce, operandy, komentáře
☐ c. instrukce, operandu, direktivy

21 S záznam slouží k

3 Pokusů

- Odpověďt: ☐ a. v případě mikropočítače HC11 není nutno tento formát používat
☐ b. přenesení zdrojového programu do prostředí vhodného pro editaci a překlad
☐ c. přenesení přeloženého programu do paměti mikropočítače

22 V okamžicích spouštění aplikace pod RTOS je před pozastavením inicializačního procesu nutno spustit

3 Pokusů

- Odpověďt: ☐ a. všechny periodicky spouštěné procesy
☐ b. minimálně jeden proces
☐ c. všechny procesy

23 Rozdíl mezi sériovou a paralelní komunikací spočívá v

3 Pokusů

- Odpověďt: ☐ a. technickém způsobu přenosu dat
☐ b. způsobu zabezpečení správnosti přenosu dat
☐ c. programovém způsobu přenosu dat

24 Na co všechno může čekající proces čekat?

3 Pokusů

- Odpověď: ☐ a. na zaplnění schránky
☐ b. na čas
☐ c. na přidělení procesoru

25 Mikropočítačový systém zahrnuje :

3 Pokusů

- Odpověď: ☐ a. Mikroprocesor, paměť, komunikační obvody, systémový software
☐ b. Mikroprocesor, paměť, komunikační obvody
☐ c. Aritmetickou jednotku, registry, řídicí obvody

26 K čemu se nedá použít schránka a zpráva?

3 Pokusů

- Odpověď: ☐ a. předání hodnoty mezi procesem a globální datovou strukturou
☐ b. předání datového pole mezi procesy
☐ c. zablokování vstupu do kritické části programu, který je využíván více procesy

27 Direktivy realizuje

3 Pokusů

- Odpověď: ☐ a. procesor
☐ b. program
☐ c. překladač

28 Pro práci s vektory a maticemi přerušení je nejvýhodnější způsob adresování

3 Pokusů

- Odpověď: ☐ a. rozšířené
☐ b. indexové
☐ c. relativní

29 Registr SP obsahuje ?

3 Pokusů

- Odpověď: ☐ a. adresu vrcholu zásobníku
☐ b. adresu dna zásobníku
☐ c. velikost zásobníku

30

Proces se může nacházet ve stavu

3 Pokusů

Odpovědět: ☐ a. čekající na prioritu☐ b. znovuspuštěný☐ c. připravený[Uložit mé odpovědi](#)Jste přihlášení jako [Novosadova Lenka](#) ([Odhlásit se](#))[MP](#)