

Technická Univerzita v Liberci

Technická Univerzita v Liberci

Hospodářská fakulta

Obor Podnikatelská Informatika



INFORMAČNÍ SYSTÉMY HOSPODÁŘSKÉ FAKULTY

BAKALÁŘSKÁ PRÁCE

Název práce:

Pracovní

Podpis:

Podpis:

Datum odevzdání:

Mgr. M. A. POKORÁ, Katedra Informatické

Pracovní, Katedra Informatické

30

1

21. 5. 1997

1997

Jakub Steiner

Technická Universita v Liberci

Hospodářská fakulta

Obor Podnikatelská Informatika

INFORMAČNÍ SYSTÉM HOSPODÁŘSKÉ FAKULTY

HF - 320 - 713

Jakub Steiner

Vedoucí práce :	Mgr. Jiří A. Randus, katedra informatiky
Konzultant :	RNDr. Pavel Satrapa, katedra informatiky
Počet stran :	39
Počet příloh :	1
Datum odevzdání :	28. 5. 1997

TECHNICKÁ UNIVERZITA V LIBERCI

Ekonomická fakulta

Katedra informatiky

Liberec, rok 1996/97

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Liberec, 28. 5. 1997

Technická univerzita v Liberci

Místopřísežně prohlašuji, že jsem bakalářskou práci vypracoval samostatně s použitím uvedené literatury pod vedením vedoucího a konzultanta.

V Liberci dne 28. 5. 1997

Seimay

TECHNICKÁ UNIVERZITA V LIBERCI

Hospodářská fakulta

Katedra informatiky

Školní rok 1996/97

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

pro

Jakub Steiner

obor č. : 62 - 53 - 7

Podnikatelská informatika

Vedoucí katedry Vám ve smyslu zákona č. 172/1990 Sb o vysokých školách a navazujících předpisů určuje tuto bakalářskou práci :

Název tématu :

Informační systém Hospodářské fakulty TUL

Zásady pro vypracování :

- 1) WWW stránky - informace o katedrách
- 2) převedení informací o fakultě ze stávajícího inf. systému
- 3) bulletin board systém
- 4) grafická úprava
- 5) další možnosti Intranetu

Seznam zkratek

ASCII	American Standard Code for Information Interchange – standard pro kódování dokumentů pro anglicky mluvící země. Definuje sadu znaků používaných v těchto zemích.
CGI	Common Gateway Interface - rozhraní mezi WWW serverem a programem nebo skriptem běžícím na tomto serveru. Vnáší možnost interaktivity do WWW.
Compiler	Překladač zdrojového kódu - program který převádí zdrojový kód programu do formy, které rozumí daný operační systém - poté lze program spustit.
FAQ	Frequently Asked Questions - často kladené otázky
Font	řez písma.
HTML	Hypertext Markup Language - standard pro formátování hypertextu používaném na WWW.
HTTP	Hypertext Transfer Protokol - protokol zabezpečující fungování WWW.
Internet	celosvětová síť fungující na protokolu TCP/IP
Intranet	síť typu LAN nebo WAN řešena pomocí internetovského protokolu TCP/IP. Má tu výhodu, že je platformově nezávislá (mohu mít ve firmě více druhů počítačů - Mac, PC, Sun ..., které by jinak těžko mohly komunikovat) a navíc mám do budoucna možnost bezproblémového napojení své sítě do celosvětového Internetu.
ISP	Internet Service Provider - Poskytovatel Internetu, firma která za úplatu umožňuje uživatelům přístup k jejímu rameni sítě Internet. Jedná se většinou o Dial-Up připojení (tel. linkou)
Java applet	Předkompilovaný program v jazyce Java.
Javascript	Netscape do svého prohlížeče implementoval skriptový jazyk nesoucí jisté shodné prvky s programovacím jazykem Java.
MIME	Multipurpose Internet Mail Extensions - standard pro formát těla elektronického dopisu. Byl zaveden z důvodů znakových specifik různých jazyků. Jelikož může být obsahem i jiný typ dat než text, používá i MIME pro data různé typy. Právě toto rozlišování typů je ve velké míře používáno i v oblasti WWW.
MS	zkratka firmy Microsoft
Popup Menu	Menu, které se po stisku zpravidla pravého tlačítka objeví na místě kurzoru. Pro programátora je použití vhodné zejména proto, že lze obsah toho menu měnit na základě polohy kurzoru.
RFC	Request for Comments - žádost o komentář. Internet nabízí novou formu spolupráce lidí na nových standardech a technologiích. Tyto dokumenty fungují jako standardní dokumentace v Internetu.
TCP/IP	komunikační a adresovací protokol používaný pro Internet. Velmi šikovným řešením ztráty dat po cestě od jednoho počítače ke druhému je rozkouskování dat na pakety a nekontrolování korektnosti během transportu, nýbrž teprve při přebírání cílovým počítačem. Pakety mohou dorazit i v jiném pořadí a naprosto odlišnými cestami. Díky obrovské propojenosti vlastního principu fungování je síť odolná na výpadky jednotlivých uzlů.
URL	Uniform Resource Locators - standard adresování v Internetu. Podporovány jsou mimo jiné např. protokoly HTTP, FTP, Gopher ...
Webmaster	Správce stránek WWW
Wildcard	„Žolíkový znak“ – speciální znak, který má speciální funkci. Např. znak „*“ všeobecně funguje jako náhražka za jakékoli slovo.

Obsah:

1	Úvod.....	6
2	Cíl bakalářské práce.....	6
2.1	Obsah	7
2.2	Forma	8
3	Stávající informační systém na Hospodářské fakultě.....	9
3.1	Infosystém	9
3.2	Nástěnky v programu Pegasus Mail	9
4	WWW stránky HF TU	10
4.1	HTML	11
4.2	URL.....	12
4.3	Základní struktura HTML dokumentu	12
4.4	Mnoho dokumentů na ploše prohlížeče	13
4.5	Rámy	14
4.6	Program pro vytváření rolovacího menu na stránkách WWW.....	17
4.7	Čeština a WWW	21
5	Struktura WWW stránek.....	23
5.1	Obecné Informace	23
5.2	Informace o studiu.....	23
5.3	Katedry	24
5.4	Rozvrhy.....	25
5.4.1	Tabulky.....	25
5.4.2	Program Převodník rozvrhů	27
5.5	Nástěnka HF.....	30
5.5.1	CGI.....	30
5.5.2	(Ne)Bezpečnost CGI skriptů	33
5.5.3	Perl	34
5.5.4	Skript.....	34
5.5.5	Alternativní řešení.....	36
5.6	Vyhledávací služby	36
5.6.1	PH Server	36
5.7	Další možnosti Intranetu.....	37
6	Závěr	39
7	Literatura	40
8	Přílohy:	41



1 Úvod

Ačkoli není Internet tak nový, jak se nám to snaží některé články v novinách a časopisech namluvit, rozhodně dosahuje v dnešní době obrovského rozmachu. Je to zejména díky relativně nové službě – WWW. WWW neboli World Wide Web je populární zejména proto, že umožňuje získávání informací z Internetu ve velmi přehledné podobě a lehce se ve WWW můžeme orientovat. Základem této služby je hypertextový dokument.

Hypertext je formát textu, který v sobě obsahuje odkazy na jiné části textu nebo dokonce odkaz na jiný dokument. Ačkoli se jedná o text, může být odkaz směřován třeba na obrázek, nebo zvuk.

Pro WWW byl definován formát hypertextového dokumentu HTML. [1] HTML je v podstatě obyčejný ASCII text, který v sobě obsahuje značky definující formátování (tučné písmo apod.), anebo právě odkazy na jiné dokumenty. K tomu, aby se text naformátoval do podoby hypertextu je potřeba ho otevřít v tzv. WWW prohlížeči. Mezi nejznámější patří např. produkt firmy Netscape – Navigator. Blíže se standardem HTML na straně 11 zabývá kapitola 4.1 - HTML.

2 Cíl bakalářské práce

Předmětem mé bakalářské práce bylo doplnění existujícího informačního systému hospodářské fakulty o prezentaci hospodářské fakulty na WWW. Ačkoli jsem měl určité znalosti v oblasti databází spojených s WWW, bohužel se jednalo o platformu MS Windows, která je na naší univerzitní půdě neaplikovatelná z důvodů všeobecné nedůvěry vůči produktům firmy Microsoft a v porovnání s řešením na platformě Linux i z důvodů finanční náročnosti. Vesměs jsem se snažil, aby byla stránka obsahem přitažlivá jak pro interní uživatele (studenty, pedagogy), tak i pro eventuelní zájemce o studium a další, kdo projeví zájem dozvědět se o



hospodářské fakultě víc. Během vypracování této bakalářské práce se stránka vyvíjela a počítám, že její vývoj neskončí po dopsání této práce.

Pouhá statická WWW stránka těžko může nosit jméno informační systém. Díky nástěnkám, PH serveru a dalším vlastnostem se však jedná o WWW prezentaci interaktivní, která slouží nejenom jako statický zdroj informací, ale i jako prostředek komunikace mezi studenty a fakultou nebo mezi jednotlivými zaměstnanci fakulty.

2.1 Obsah

Obsah WWW stránek čerpá z velké části z již existujících programů, které jsou ale dostupné pouze pro interní uživatele a netvoří komplexní celek s jednotným ovládáním. Ve stručnosti zde uvedu několik druhů dat, které lze na WWW stránkách nalézt:

- **Obecné informace o fakultě** – Sídlo, vznik, vedení fakulty, studijní obory, počty uchazečů o studium, systém studia a další všeobecné informace
- **Informace o studiu** – Podrobné informace o přijímacím řízení, studijních oborech, studijních plánech. Dále jsou k dispozici směrnice k vypracování bakalářských a inženýrských prací, studijní praxe a závěrečných zkoušek. Velice rozsáhlou částí jsou **sylaby předmětů** vyučovaných na HF.
- **Informace o katedrách** – Stručné informace o jednotlivých katedrách na HF. Použitá data byla získána od pedagogů pomocí elektronického dotazníku.
- **Rozvrhy studijních skupin a učeben** – Každý student má k dispozici rozvrh své studijní skupiny. Některé vytížené učebny zde taktéž mají podobný rozvrh. Všechny tyto dokumenty jsou automaticky generovány.
- **Nástěnka** – Každý student nebo pedagog má možnost „vyvěsit“ na této elektronické nástěnce zprávu určenou pro větší okruh studentů, či pedagogů.



- **Vyhledávací služba PH serveru** – Tato služba umožňuje vyhledávat stručné informace (e-mail, telefon) o studentovi, či zaměstnanci HF, na základě jména či příjmení.

2.2 Forma

Při publikování nové WWW stránky, která by měla být jádrem informačního systému hospodářské fakulty Technické University v Liberci jsem vycházel z následujících zkušeností :

- Stránka by měla být přístupná i pro majitele modemů. Ačkoli není snadné udělat stránku přitažlivou bez přítomnosti obrázků, musí existovat možnost prohlížet stránku v čitelné podobě i při vypnutí nahrávání obrázků na straně klienta
- Stránka by měla být viditelná i na prohlížečích starších. Pro kontrolu vzhledu na různých prohlížečích jsem na svém počítači nainstaloval víc prohlížečů, než by kdekomu připadalo přiměřené. Stránky HF TU jsem testoval na 32 bitových verzích prohlížečů Netscape Navigator 3 pro Windows 95 a Linux, Netscape Navigator 4 pro Windows 95, MS Internet Explorer 3.02 pro Windows 95 a NCSA Mosaic 3 taktéž pro Windows 95. Ze 16 bitových verzí mi při testování postačil Netscape Navigator 1.4.
- Stránka by měla být dobře strukturována jak z pohledu lepší orientace na ní, tak z pohledu pozdějších úprav. Abych mohl splnit tento a předchozí požadavek, bylo nutné vytvořit dvě verze WWW stránek. (viz podrobnější informace v kapitole 4.5 - Rámy, strana 14)
- Vlastní vytvoření stránky ještě neznamená, že bude nalezena v džungli WWW. Dalším krokem je zaregistrování do mnoha WWW indexů.



3 Stávající informační systém na Hospodářské fakultě

3.1 Infosystém

Infosystém je program umožňující prohlížení dat o Technické Universitě v prostředí operačního systému MS DOS. Data v něm obsažená byla hlavním zdrojem informací nyní dostupných na WWW stránkách hospodářské fakulty.

Vnitřní síť budov Technické University je řešena na software Netware 3 firmy Novell. Na této platformě byl také vytvořen systém umožňující hierarchizované umisťování informací a to hned několika katedrám. Netware totiž umožňuje uživatele řadit do skupin a těm pak dávat specifická práva přístupu k jednotlivým adresářům. Jde tedy umožnit všem uživatelům číst zprávy ze všech adresářů a zároveň umožnit některým uživatelům zapisovat do vybraných adresářů, a tím jim umožnit přidat do Infosystému další informace nebo změnit stávající. Vlastní „prohlížeč“ nebo klient, se kterým uživatel pak pracuje, mu umožňuje listovat v hierarchickém menu s odkazy na další menu, nebo přímo na dokument. To, že se v tu chvíli pohybuje po adresářové struktuře, uživatel ani neví. Vlastní dokumenty příspěvků jsou ve formátu t602, jelikož v době tvorby tohoto Infosystému byl nejpoužívanějším textovým editorem právě tento produkt firmy Software 602.

3.2 Nástěnky v programu Pegasus Mail

Velice dobrým a často používaným klientem pro elektronickou poštu je program Pegasus Mail. Kromě běžných funkcí spojených s psaním a vedením elektronické korespondence umožňuje zároveň na lokální síti vést takzvané nástěnky (*noticeboard*). Jedná se o elektronickou formu klasické nástěnky. Jedná se tedy o místo, kde různí lidé mohou dávat na vědomí zprávy, které uznají za vhodné. Tato elektronická forma zároveň umožňuje zprávy tematicky rozdělit, takže je ve zprávách lepší orientace. Podobně jako na klasickou nástěnku však nemůže každý „připíchnout“ cokoli se mu zlíbí.

Podobně jako v Infosystému řeší Pegasus přístupová práva pomocí přístupových práv vlastního síťového software Netware. (viz kapitola 3.1 - Infosystém, strana 9) Každému uživateli nebo skupině je možné přidělit přístupová práva k adresáři na diskovém svazku serveru. Mezi základní práva patří právo číst, zapisovat či mazat soubory v adresáři. Pegasus kontroluje podle indexového souboru který vytváří, jestli má uživatel právo do nástěnky (adresáře) zapisovat a teprve potom mu umožní vytvořit zprávu. Tento soubor je však pouze informativní záležitostí pro Pegasus Mail, rozhodující je nastavení v Netware. Vlastní zprávy jsou pak vytvářeny přímo jako ASCII soubory v daném adresáři. Pro potřeby české diakritiky je použito kódování MIME.

4 WWW stránky HF TU

WWW stránky hospodářské fakulty jsem koncipoval ze dvou základních pohledů:

- Jako WWW prezentace pro eventuální zájemce o studium. Svou povahou to jsou externí uživatelé.
- WWW prezentace pro studenty a pedagogy Hospodářské fakulty. Do budoucna (např. na poli BBS hospodářské fakulty - viz kapitola 5.5 - Nástěnka HF, strana 30) by některé informace nebyly přístupné veřejně, ale jen pro interní uživatele.

Většina dat však bude svou povahou pro oba typy uživatelů. Všeobecné informace o studiu budou zajímavé spíše pro externí uživatele, ale např. informace o vyučovaných předmětech budou hodnotné pro obě skupiny. Než se ale pustím do obsahu, rád bych vysvětlil několik pojmů a technik, které jsou nutné pro pochopení procesu vytváření WWW stránek. Co obnáší taková tvorba WWW stránek, jak je okořenit interaktivitou, jak automatizovat některé postupy? Na podobné otázky se pokusím odpovědět na základě svých zkušeností s tvorbou WWW dokumentů.

4.1 HTML

Termín HTML je i mezi laiky velmi známý. Jedná se o formátovací jazyk, který je využíván právě pro zobrazování dokumentů na WWW [1]. Dokument, jež je klasickým ASCII textem obsahuje speciální značky - příkazy které definují, že například následující text bude nadpisem, nebo skokem na jiný dokument. „Po drátech“ je přenášen jako klasický text, teprve u klienta (WWW prohlížeče, browseru) je formátován do podoby, kterou známe. To s sebou přináší jak výhodu neomezené přenositelnosti, (existuje mnoho klientů pro mnoho operačních systémů a každý má možnost vidět ten samý dokument, jelikož formát obyčejného textu je čitelný na všech platformách) tak zároveň nevýhodu v tom, že na různých prohlížečích může HTML dokument vypadat trochu jinak. Tuto různost lze do jisté míry eliminovat tím, že se striktně budu držet standardů HTML. Ale již tato věta nemá pro toho, kdo zná HTML příliš velkou vypovídací hodnotu. Standardů je totiž na tomto poli mnoho (poslední verze HTML je verze 3.X) a navíc většina prohlížečů z důvodů lepší konkurenceschopnosti přidává vlastní značky, kterým leckterý webmaster neodolá a pak je problém na světě. Naštěstí existuje pravidlo, že v případě, že prohlížeč značku nezná, ji bude ignorovat. Tak je možné aplikovat značku specifickou pro jeden prohlížeč, zároveň ale nabídnout verzi pro prohlížeče, které jí nerozumí. Nestandardní značky totiž často obsahují pasáž, kam je možné vložit alternativní text pro starší prohlížeče, který je zas ignorován prohlížeči podporující tuto nestandardní značku.

V současné době je v médiích připomínána dvojice soků na poli WWW prohlížečů - Navigator firmy Netscape a Internet Explorer firmy Microsoft. Mám tu zkušenost, že většinu rozdílností lze řešit použitím obou alternativ, jak pro IE, tak Netscape. Jestliže je stránka správně formátovaná na těchto dvou prohlížečích, má webmaster zhruba 90% pravděpodobnost, že ji tak uvidí všichni uživatelé, neboť jen málokdo používá prohlížeč jiný. (stránka HF TU je čitelná i na prohlížeči NCSA Mosaic, prohlížeči který se významnou mírou zasloužil o obrovskou popularitu WWW)

4.2 URL

URL neboli Uniform Resource Locator je způsob jak explicitně vyjádřit adresu dokumentu v Internetu. [2] Ačkoli je velkou měrou využíván pro protokol přenosu hypertextových dokumentů (HTTP), je možné pomocí URL specifikovat i dokumenty při použití jiných přenosových protokolů. Mezi ty základní patří FTP (file transfer protocol), Telnet, Gopher a News. Spektrum protokolů však není těmito uvedenými vyčerpáno.

V době výstavby WWW stránek hospodářské fakulty, měla úvodní stránka URL <http://www.kin.vslib.cz/~steiner/tul/index.html>

4.3 Základní struktura HTML dokumentu

Každý HTML dokument se skládá z vlastního textu a značek definujících, jak tento dokument má vypadat, eventuálně do něj je možné těmito značkami vložit obrázek, 3D scénu, zvuk či cokoli jiného. Vlastní dokument by podle HTML 2 měl mít následující strukturu:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<HTML>
<HEAD>
  <TITLE>titulek</TITLE>
</HEAD>
<BODY>
  tělo dokumentu
</BODY>
</HTML>
```

Ve značkách není třeba dbát na velikost písmen (malá/velká) ani není třeba dbát výše uvedeného zarovnání. V případě, že bych chtěl, mohu celý HTML dokument napsat do jedné řádky. Nový řádek se ve výsledném naformátovaném dokumentu neukáže, pokud nepoužijete značku nového odstavce <p>, nebo značku pro zlomení řádky
. Je to z důvodů platformové nezávislosti HTML dokumentů.

Různé operační systémy končí řádku různými znaky, takže by se dokument na různých operačních systémech ukazoval jinak.

Ted' ale zpět ke struktuře HTML dokumentu. Základním prvkem, je párování značek. Ačkoli existují značky, které lze použít samotné má vesměs každá značka `<značka>` svou kolegyni `</značka>`, která vyjadřuje, že platnost funkce dané značkou ted' končí. Značky `<html></html>` označují začátek a konec HTML dokumentu. První řádek vysvětluje, o jaký dokument se jedná. Ve většině prohlížečů platí, že vše mimo značky `<html></html>` je ignorováno, nicméně se může stát, že syntaxi porušíte a přitom vám ji váš prohlížeč zobrazí správně. Z těchto důvodů existují různé korektory syntaxe, které vás na podobné nesrovnalosti upozorní. Moje zkušenost je však taková, že se jedná o kontrolu podle standardu HTML 2 a mnoho již dnes základních značek jako třeba tabulky prostě nekontrolují. Vlastní tělo dokumentu se nachází mezi značkami `<body>` a `</body>`. Nebudu se zde rozepisovat o všech značkách standardu HTML verze 2, nicméně o jedné se přeci jen zmíním. Jedná se značku, která se stará o to, že je hypertext HTML hypertextem, tedy dokumentem s odkazy na jiné dokumenty. Pomocí značky `<a></a>` je možné definovat text, nebo jiný objekt, ležící mezi těmito značkami, který je citlivý na uživatelské poklepnutí myší. Pak je prohlížeč nasměrován na nové URL, které je v rámci značky `<a>` definováno parametrem `href`. Například `<a href="dalsidokument.html">odkaz na další dokument</a>` vytvoří text odkaz na další dokument. Jakmile na tento text klepnete myší, přenesete Vás prohlížeč na dokument `dalsidokument.html`.

4.4 Mnoho dokumentů na ploše prohlížeče

Orientace po WWW stránkách je často velmi nesnadná. Člověk je zavalen množstvím odkazů a pak je těžké se vrátit zpět na určité místo a pokračovat jinou cestou. Základem WWW je hypertext, tedy text obsahující skoky na jiná místa v dokumentu, nebo dokonce odkazy k dokumentům na druhém konci světa.

Z pohledu administrátora WWW stránky je pro mne však důležité, aby se v datech, která nabízím, šlo snadno orientovat, a aby uživatel viděl vše, co bylo mým záměrem. V případě, že je od svého původního záměru odkloněn díky nějakému v tu chvíli zajímavému odkazu, nemá pak šanci se vrátit jinak, než že se vrací tou samou cestou, přes všechny odkazy, které uskutečnil. Často se pak stává, že již dávno zapomněl, proč že to vlastně tráví čas na vaší stránce. Teprve v kombinaci s nějakou strukturou je hypertext ideální formou podání psaných dat, ze kterých je možné si vybrat jen to důležité.

4.5 Rámy

Dnešní prohlížeče jako je např. Netscape Navigator a Internet Explorer již mají implementovány značky, které umožňují zobrazení několika HTML dokumentů najednou. Laikovi by mohlo připadat, že to vnáší ještě větší faktor nepřehlednosti, než když se uživatel bude motat v odkazech jednoho dokumentu. Opak je pravdou. Základním předpokladem zpřehlednění užitím těchto oken je vzájemná souvztažnost jednotlivých dokumentů. Odkazy z jednoho dokumentu jsou otevírány v okně druhém, neztrácím tedy dokument, ze kterého jsem odkaz spustil a mohu se tedy snadno vrátit.

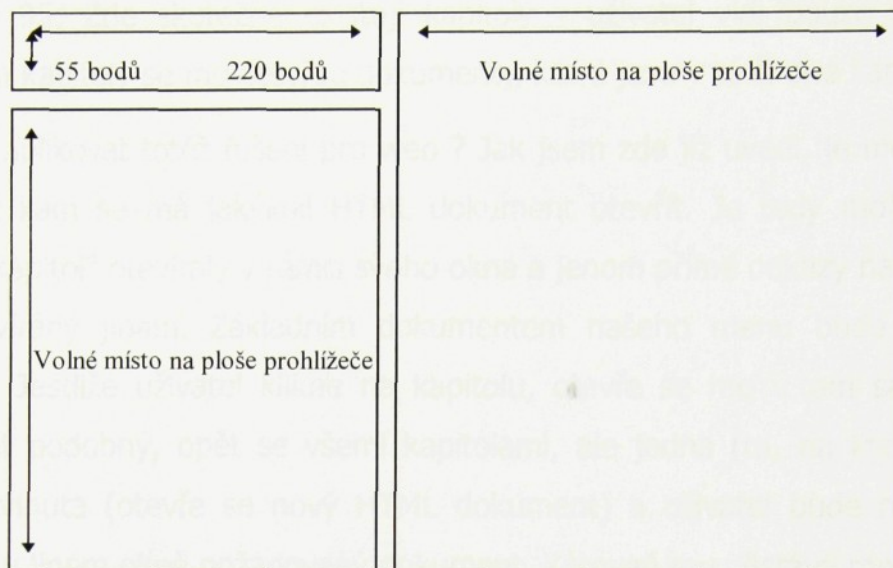
Jak tedy rámy fungují ? Rozvržení oken po pracovní ploše prohlížeče je řešeno značkami `<frame>` a `<frameset>`. [4] `<frame>` udává, jaký dokument má být uložen v okně s určitým názvem. Jméno okna je pak používáno při adresování místa, kam má být odkaz otevřen. `<frameset>` se používá tam, kde chci určitým způsobem rozvrhnout na obrazovce více sloupců/řádků oken a definovat jejich šířku/délku. Šířku či délku lze definovat v bodech a jelikož může mít klient různé rozlišení, dá se použít i „žolíková značka“ „*“, která zde funguje jako zbytek místa na ploše prohlížeče. Dále lze ve značce `<frame>` definovat scrollování textu - jestli se v případě, že se text nebo obrázek nevejde do okna, mají objevit scrollovací lišty, dále pak jestli v okně budou okraje, jestli bude možné okno uživatelem zvětšit nebo zmenšit.

Zvláštností jsou třeba definování tloušťky a vystoupení hranic oken, které nejsou popsány ve vlastním standardu, a proto si je např. Microsoft i Netscape ošetřil vlastními parametry ve značkách `<frame>` a `<frameset>`. Každý jinak, pochopitelně. Naštěstí je možné je zapsat do značky oba, takže dosáhnete toho, že Navigator i Internet Explorer zformátují dokument stejně, i když každý podle jiných parametrů.

V poslední řadě se zde ještě zmíním o značce `<noframes></noframes>`. Jak jsem se zmínil na začátku, umí rámy ukázat pouze prohlížeče, které mají implementovány tento „standard“, implementovaný poprvé firmou Netscape. Jelikož však po světě existuje ještě mnoho WWW prohlížečů z doby kamenné, nebylo by vhodné lidem používajícím prohlížeče jiných společností než Netscape a Microsoft znemožnit přístup k našim datům. Právě o to se stará značka `<noframes>` vše co se nachází uvnitř této značky (rozuměj `<noframes></noframes>`) bude zformátováno pouze na prohlížečích neovládajících rámy. (`noframes` pouze zamezuje, aby text byl ignorován na prohlížečích, které rámy umí - starší prohlížeč značky, kterým nerozumí, jednoduše přeskakuje. To znamená že přeskočí všechny značky `<frame>`, `<frameset>` i `<noframes>` a formátuje klasický HTML dokument. Pro příklad uvedu, jak je rozvržena obrazovka pro WWW hospodářské fakulty:

```
<frameset cols="210,*" framespacing="0" frameborder="0"
border="0">
  <frameset rows="55,*">
    <frame src="maly_dokument.html" name="logo"
marginwidth="0" marginheight="0"
scrolling="No" noresize>
    <frame src="odkazovy_dokument.htm"
name="odkazy" marginwidth="4"
marginheight="4" scrolling="Auto">
  </frameset>
  <frame src="hlavni_dokument.html" name="hlavni"
scrolling="Auto">
</frameset>
```

naformátuje následující obrazec:



Obrázek 1: Plocha prohlížeče je značkou `<frameset>` nejprve rozdělena na dva sloupce, a v prvním sloupci se pak nachází další rozdělení pomocí druhého `<frameset>` na dva řádky.

Dalšího členění lze dosáhnout, když do nějakého okna otevřeme dokument obsahující další značky pro formátování rámců.

Jak je ale řešeno zobrazování odkazů ve správných oknech? Řekněme že bychom například chtěli, aby se odkaz z okna „odkazy“ zobrazil v okně „hlavni“. Definici cíle lze upřesnit pomocí parametru `target` ve vlastním odkazové značce `<a>`. V našem případě tedy `<a href="nejaky_dokument.html" target=hlavni> Odkaz </a>`. Dokonce v tomto případě, kdy z jednoho okna směřuje většina odkazů právě do okna „hlavni“, je možné si práci zjednodušit vložení parametru `target` do značky `base` (`<base target="hlavni">`) a v případě, že `target` ve značce `<a>` nespecifikujeme, bude implicitně použito „hlavni“.

Díky těmto značkám máme tedy možnost umístění určité kostry, nebo obsahu do jednoho dokumentu, aby měl uživatel možnost se vždy vrátit k nějaké jiné „kapitole“ bez ztráty možnosti pohybovat se po dokumentu tak, jak byl zvyklý dříve. V případě, že bychom ale měli takovýchto „kapitol“ mnoho, by se přehlednost toho menu výrazně zmenšila. Při hledání řešení mne napadlo, jak tento problém řeší např. index nápovědy k aplikacím v operačním systému

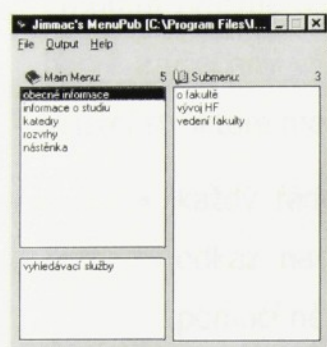
Windows 95. Zde skutečně existují kapitoly - uživatel vidí pouze je a teprve otevřením kapitoly se mu otevrou dokumenty, které jsou v té či oné kapitole.

Jak tedy aplikovat totéž řešení pro web ? Jak jsem zde již uvedl, je možné přesně definovat kam se má jakýkoli HTML dokument otevřít. Je tedy možné, aby se odkazy „kapitol“ otvíraly v rámci svého okna a jenom přímé odkazy na dokumenty byly otvírány jinač. Základním dokumentem našeho menu bude tedy výpis „kapitol“. Jestliže uživatel klikne na kapitolu, otevře se mu v tom samém okně dokument podobný, opět se všemi kapitolami, ale jedna (ta, na kterou kliknul) bude rozvinuta (otevře se nový HTML dokument) a uživatel bude mít možnost otevřít si v jiném okně požadovaný dokument. Zároveň mu zůstává možnost přímo kliknout na jinou kapitolu, nemusí se tedy nikam vracet. V případě statické WWW stránky by bylo možné všechny dokumenty snadno ručně vytvořit. Statická stránka však ztrácí rychle svou informační hodnotu a většina měnících se informací mění zároveň i svou strukturu. Např. při přidání, nebo změně kapitoly, bych nebyl nucen editovat jen jeden dokument, nýbrž hned $1 + \text{počet kapitol}$ dokumentů. Právě proto jsem se rozhodl vytvořit vlastní nástroj k tvorbě podobného menu.

4.6 Program pro vytváření rolovacího menu na stránkách WWW

V době, kdy jsem byl postaven před tento problém, jsem neovládal nijak obstojně žádný programovací jazyk. Jednoduché programky v Pascalu jsem sice již programoval, ale na program pro vytváření a údržbu takového menu včetně výstupu v HTML jsem si v Pascalu netroufal hlavně z důvodů časové tísně. Musel bych nejdříve vytvořit nějaké uživatelské rozhraní, a s tím jsem pod Pascalem neměl žádné zkušenosti.

Firma Microsoft je známa mnoha svými nestabilními produkty doplněnými smrtícím marketingem. O jazyku Visual Basic však mohu říct, že mě příjemně překvapil, protože splnil to, co sliboval. Pomocí něj může jakýkoli začátečník vytvořit jednodušší aplikaci (neberme pod pojmem jednodušší aplikace program typu „hello world“, v pojetí VB je jednoduchou aplikací například i program pracující s relačními databázemi) bez nutnosti programovat vlastní prostředí. To ve VB doslova „kreslíme“. Výsledkem jsou uživatelsky příjemné a z hlediska rozhraní již



Obrázek 2 Uživatelské rozhraní programu MenuPub

ošetřené aplikace.

Prvním krokem bylo tedy nadefinovat (či nakreslit) prostředí programu. Chtěl jsem, aby k používání programu nebyl nutný jakýkoliv manuál, což je např. další výhoda programů firmy Microsoft. I tak kritizovaný a zastaralý operační systém jako je MS DOS lze nainstalovat bez jediné stránky manuálu a jakékoli předchozí zkušenosti. Zkuste podobným způsobem nainstalovat Linux. U Linuxu se musíte nejdřív probrat množstvím textu, abyste se vůbec dozvěděli, co k instalaci potřebujete a co ne.

Vlastní prostředí podlehlo několika minoritním změnám, ale v podstatě vypadalo tak, jak je vidět na obrázku.

Plocha aplikace je rozdělena do tří částí. První (vlevo nahoře) je ta část, kde jsou umístěny položky menu, které budou fungovat jako „kapitoly“. K nim jsou souvztažně přiřazeny položky druhého menu - menu každé kapitoly (vpravo). Při vybrání jedné položky v části kapitol se seznam vpravo naplní daty příslušícími dané kapitole. Ve spodní levé části je ještě seznam přímých odkazů, tedy odkazů nezařazených v žádné kapitole.

Hlavní menu se sestává pouze ze základních položek typu otevři soubor, ulož soubor a nový. Dále je zde položka nastavení, kde uživatel určí konečný vzhled menu - barvu pozadí, ev. obrázek na pozadí, barvu textu, odkazů a již viděného

odkazu. Dále je tu jméno okna, kam se mají implicitně otevírat dokumenty z odkazů v menu (viz. `<base target=>` v kapitole 4.5 - Rámy, strana 14).

Zbytek příkazů se zadává z tzv. popup menu. Toto menu se objevuje po zmáčknutí pravého tlačítka a jeho obsah se mění na základě oblasti, kde se právě nachází kurzor. Toto menu obsahuje položky typu přidej položku, odeber položku, vyměň položku, přejmenuj položku a uprav URL položky.

Program pracuje s textovými soubory .but, ve kterých si ukládá jak informace o barvě a všechny výše uvedené vlastnosti HTML dokumentu a hlavně informace o vlastní struktuře menu a jednotlivých odkazů v nich. Formát je následující:

- každý řádek začíná číslem vyjadřujícím buď číslo kapitoly, ve které se odkaz nachází (u vlastních odkazů), nebo číslo vyšší, vyjadřující kód pomocí něhož lze určit o jakou položku se jedná.
- položky na řádce jsou oddělovány speciálním znakem, který se těžko objeví v jakémkoli smysluplném textu.
- na prvních devíti řádcích jsou informace o úpravě dokumentu. (barvy atd.) Tyto řádky začínají číslem 999.
- dalšími položkami jsou kapitoly, kde je uveden po čísle 256 (může existovat pouze 256 kapitol - číslováno od 0) jenom název kapitoly.
- číslem 666 začínají řádky s informacemi o přímých odkazech. Jsou to ty odkazy, které se nenacházejí v žádné kapitole a jsou přístupné rovnou z hlavního menu. Uvedeno je jak jméno odkazu, tak URL
- poslední kategorií jsou vlastní odkazy zařazené v kapitolách. Maximálně jich může být 256 v každé z 256 kapitol. Na první pozici je číslo vyjadřující příslušnost ke kapitole. Formát nedovoluje přeskupené odkazy v různých kapitolách. (Nejde mít např. první dva odkazy z první kapitoly a další tři z další a pak opět z první... při vytváření souboru přímou editací je toto nutné mít na paměti, neboť se pak menu správně nevygeneruje a

program vynechá odkazy z opakující se kapitoly. Není sice problém toto programově vyřešit, ale jelikož jsem byl nucen čas využít na jinou činnost a vlastně nebyla ani potřeba manuálně se soubory pracovat - program sám je generuje správně, nechal jsem tuto možnost otevřenou do další verze programu.) Další položky jsou stejné jako u přímých odkazů, tedy název odkazu a URL.

Uživatel má možnost vytvořit si vlastní menu, a pro pozdější úpravy jej uložit ve formátu .but (popsaným výše). Hlavní funkcí programu je ale generování HTML stránek. K tomuto účelu může uživatel specifikovat adresář, kam se menu v HTML vygeneruje. Je vhodné mít pro menu vlastní adresář aby měl uživatel jistotu, že žádný soubor, který v adresáři existuje, nebude přepsán souborem z menu. Soubor hlavního menu má název `buttons.htm` a všechny další jsou generovány podle jména kapitoly. Při vytváření názvu je tedy třeba ošetřit následující situace:

- Program podporuje češtinu v kódu 1250. (viz kapitola 4.7 - Čeština a WWW, strana 21) Z hlediska mnoha platforem, kde může být soubor umístěn (server na bázi Windows je v našich zemích zatím (naštěstí?) raritou, většina serverů pracuje na operačních systémech UNIX) není dobré, aby název obsahoval některé znaky specifické pro češtinu. Všechny znaky tohoto typu jsou nahrazeny jejich ekvivalenty bez znaménka (á-a ě-e š-s atd.)
- Program MenuPub je spustitelný i pod Windows 3.0, která používají souborového systému operačního systému MS-DOS, který neumožňuje jména souborů delší než 8 znaků plus tři jako přípona.
- Může se stát, že bude počáteční slovo jména kapitoly kratší než osm znaků. Po tomto slově nemusí následovat nic, což z hlediska formátu názvu souboru nevádí, nebo může následovat mezera. Mezery jsou v tomto případě nahrazeny znakem `_`.

- Dále existují další speciální znaky, které se ve jménu souboru nemohou nacházet. Jsou to zejména tzv. wildcards, které pomáhají při práci s více soubory, a další speciální symboly. (*?.;: atd.) Všechny tyto znaky jsou pak nahrazeny znakem _.

Jak jsem se již zmínil, program generuje české znaky v kódové stránce operačního systému Windows - 1250. Aby bylo možné umožnit čtení textu se správnými diakritickými znaménky i na jiných systémech, je nutné použít konverzního programu pracujícího na straně serveru. Podrobně se problematikou češtiny zabývá kapitola 3.3. MenuPub podporuje jeden takovýto program, který je v rámci ČR velmi populární - SaCzech. Slovo podporuje je možná nadnesené, neboť jediné co MenuPub ohledně češtiny dělá, je vygenerování řádku obsahujícího informaci o kódování dokumentu, kterému by SaCzech „rozuměl“. V tomto případě jde o `<!-- MYCHARSET=CP1250-->`.

4.7 Čeština a WWW

S češtinou jsem neměl problémy jenom já na základní škole. S češtinou má bohužel problémy i Internet. HTML verze 2 již sice pamatuje na různá specifika neanglických národů, nicméně pouze těch západoevropských. HTML 3 již podporuje mimo jiné i ISO 8859-2 (ISO Latin 2), standard pro střední a východní Evropu, bohužel stále se ještě pohybujeme v době „vlády“ HTML 2. Prvním předpokladem snahy nabídnout uživateli dokument se správně zobrazujícími se českými znaky je, možnost uživatele prohlížet si ve svém operačním systému české dokumenty lokálně, tedy má i český font. Jeho prohlížeč by měl být schopen tento font použít.

Pro vytváření HTML stránek se běžně používá klasického ASCII kódu. I když není na češtinu myšleno, přenáší protokol HTTP data osmibitově, je možné české znaky přeci jen poslat. Problém je ovšem v různém kódování českých znaků. To, že se mi čeština zobrazí správně lokálně, ještě neznamená že se tak stane i mému kolegovi

pracujícím na jiném operačním systému, který používá jiný standard kódování. Při tvorbě textů bychom měli myslet na všechny česky mluvící uživatele (kromě těch v zahraničí, marně shánějících český font...) a tudíž jim dát možnost vybrat si dokument kódovaný v jim srozumitelné podobě. Varianta, kdy bychom veškeré dokumenty shromažďovali v několika exemplářích různého kódování však také nezní líbivě. Variantou druhou je překódování dokumentu pomocí CGI skriptu až po dotazu klienta. Nevýhodou této metody je sice větší zátěž serveru (která je například na serveru OTUS viditelná pouhým okem ☺) oproti metodě vytváření mnoha verzí dokumentů, ale přináší s sebou obrovskou výhodu nižší náročnosti na údržbu stránek. Jelikož si už dal někdo práci s vytvářením podobných CGI skriptů a jejich relativní náročnosti a vesměs zbytečnosti vkládat úsilí do něčeho, co už funguje, rozhodl jsem se využívat CGI skriptu SaCzech, jehož autorem je RNDr. Satrapa. Je nainstalován na serveru, kde jsou umístěny stránky HF TU, takže stačilo se starat o to, abych SaCzkovi (zde si dovolím skloňovat nesklonovatelné) předložil dokument, který je identifikován řádkem obsahujícím `<!-- MYCHARSET=CP1250-->`, který mu říká, že dokument je kódován češtinou používanou v operačním systému MS Windows. Uživateli je pak nabídnuto několik kódování a dle vybraného typu jsou pak další dokumenty kódovány správně. To zabezpečují CGI skripty jejichž název je tvořen výsledným kódováním. Například pakliže chcete dokument v kódu 1250, je místo URL `http://server/~steiner/dokument.htm` uživatel nasměrován na `http://server/cgi-bin/toCP1250/~steiner/dokument.htm`. Pro vlastní výběr kódování je možné použít CGI skript *whichcode*. Na stránce HF TUL jsem se ale rozhodl pro přímé odkazy na příslušné překódovací skripty. WWW stránky HF TUL používají relativních odkazů na dokumenty, takže při pohybu po stránkách fakulty je kód zachováván. Jestliže existuje uživatel, který používá kódování češtiny, které není SaCzkem podporováno (nedokážu si žádné představit), stále má možnost vybrat si standardní ASCII kódování, při kterém jsou všechny háčky a čárky odstraněny.



5 Struktura WWW stránek

5.1 Obecné Informace

V této kapitole jsou k dispozici informace o fakultě, včetně jejího historického vývoje a informace o současném vedení fakulty. Všechny tyto informace pochází z výše zmíněného Infosystému na serveru TYTO.

K vlastnímu zpracování bych rád zmínil problém s různým zobrazením na prohlížečích Internet Explorer a Netscape Navigator - veškerý text kromě nadpisů je odsazen pomocí značky `<blockquote>`, což by mělo korespondovat s dvoubarevným pozadím. Velikost odsazení se však liší u obou prohlížečů. Musel jsem nastavit pozadí tak, aby se hodilo pro obě vzdálenosti. (zkoušel jsem pro odsazení použít i *definition list* `<dl>`, tam je problém ale ještě markantnější) Dalším řešením by bylo použití tabulek, pro větší množství textu je však toto řešení nevyhovující z následujícího důvodu: Při formátování tabulky musí prohlížeč načíst celou tabulku, aby nedošlo k chybnému zobrazení. Pro uživatele to znamená, že dokud celý kus dokumentu obsahující tabulku není načten, nezobrazí mu prohlížeč ani písmeno. Je zde možnost rozdělit text do několika tabulek, ale je to řešení velice náročné na čas a zpracování.

5.2 Informace o studiu

V další kapitole už se nachází informace důležité jak pro zájemce o studium na hospodářské fakultě TU, (přijímací řízení, studijní obory) tak i pro vlastní studenty - studijní plány všech oborů a ročníků. Dále jsou zde k dispozici směrnice pro zajištění odborné praxe a směrnice pro zpracování bakalářských prací a vykonání závěrečných bakalářských zkoušek.

Nejrozsáhlejší položkou této kapitoly jsou osnovy předmětů vyučovaných na HF. Osnovy předmětů se rovněž nacházejí v Infosystému na serveru TYTO. Bohužel jejich struktura není natolik standardizována, aby šla řešit automatickým generováním (v rámci vymezeného času), jako jsem řešil např. rozvrhy studijních

skupin a učeben. (viz kapitola 5.4 - Rozvrhy, strana 25) Jednalo se tedy o velmi stereotypní práci formátování existujících dokumentů do HTML. (Jednalo se zhruba o 100 dokumentů).

V jednotlivých sylabech předmětů je možné se dočíst informace o přednášejících, cvičících pedagogích včetně katedry, která předmět garantuje. Dále pak informace o rozsahu přednášek a cvičení, podmínky pro udělení zkoušky či zápočtu, o podpůrných prostředcích výuky, doporučené literatuře a pak vlastní osnovu předmětu.

5.3 Katedry

Informace o katedrách již nebyly na serveru TYTO k dispozici a bylo nutné rozvrhnout, o jaké informace by se mělo jednat. Podle vzoru zahraničních universit jsem společně s RnDr. Satrapou a Ing. Skrbkem vytvořil dotazník, který byl pak rozeslán jednotlivým pedagogům fakulty. Dotazník měl dvě části - část týkající se katedry a část směřovaná přímo na osobní informace pedagoga. První část obsahovala položky týkající se garantovaných předmětů, zaměření katedry, vědeckovýzkumná a hospodářská činnost apod.

Část osobních dat obsahovala mimo jiné i dotazy na funkci, odborné zaměření, jazykové znalosti, garance předmětů a publikační činnost pedagoga.

Dalším problémem byla distribuce těchto dotazníků. Pochopitelně nebylo žádoucí je distribuovat jinak než elektronicky, neboť by to s sebou přinášelo problémy spojené s opětovným přepisováním dat při vytváření HTML stránek.

Jelikož jsem chtěl, aby se na WWW stránkách kateder objevily i fotografie pedagogů, bylo nutné přijít na nejjednodušší způsob jak elektronické fotografie pořídit. Ačkoli katedra vlastní stolní barevný scanner, neexistují fotografie stejného formátu pro použití na WWW. Muselo se tedy přistoupit k fotografování. Protože nebylo nutné dosáhnout při focení vysoké kvality pořízených fotografií, postačil k fotografování digitální fotoaparát Quicktake firmy Apple. Odpadla tak zdlouhavá

část převodu fotografií do digitální formy pomocí scanneru, neboť místo filmu používá tento přístroj interní paměť do které už snímky v digitální formě ukládá. Je možné pomocí obslužného software převést fotografie do upravitelného formátu a pak po menších úpravách velikosti je možné takovéto fotografie použít přímo pro web.

5.4 Rozvrhy

Rozvrhy patří k datům, kvůli kterým je Infosystém vůbec využíván. Svou povahou jsou to velmi často se obnovující data. Z tohoto důvodu je nesmysl je formátovat z Infosystému do HTML manuálně. Pro konverzi textových tabulek bylo nutné vytvořit program, který by umožnil lehkou transformaci tabulek. Protože musí být program přístupný i pro budoucí správce WWW stránek fakulty, upustil jsem od původního záměru naprogramovat převodník opět v mnou oblíbeném Visual Basicu. Ačkoli má pro práci s řetězcí VB nesmírně silné funkce, rozhodl jsem se pro Pascal, který umožňuje zkompileovat spustitelný program v holém DOSu a zabírá nesrovnatelně méně diskového prostoru.

5.4.1 Tabulky

Tabulky jsou velmi užitečným nástrojem při formátování textu i grafiky a jsou velmi často používaným elementem každého HTML dokumentu. Pro pochopení procesu převodu obou formátů si vysvětleme princip formátování tabulek v HTML:

Každá tabulka začíná a končí značkami `<table>` a `</table>`. V počáteční značce můžeme specifikovat několik parametrů. Jedním z důležitých je šířka okrajů v tabulce. Pakliže nastavíme tento `border` parametr na hodnotu 0, nebudou okraje tabulky vidět. Toto se velmi často používá pro přesné formátování textu na obrazovce, neboť vlastní tabulka pak není vidět. Dalšími užitečnými parametry jsou šířka (`width`), která může být specifikována např. v procentech volného místa v okně prohlížeče, nebo přímo v obrazovkových bodech (pixelech). Další velmi efektní je možnost zobrazit pod tabulkou obrázek. Bohužel není tato schopnost

implementována do prohlížeče Netscape Navigator, takže není často používána, neboť Netscape vlastní zhruba 60% trhu na poli WWW prohlížečů, což vypovídá o oblíbenosti a rozšířenosti těchto prohlížečů mezi uživateli Internetu.

Prohlížeč zobrazuje tabulku teprve po přečtení poslední značky `</table>`, protože se s každým dalším řádkem tabulky může radikálně změnit vzhled tabulky. Jednotlivé řádky jsou definovány značkou `<tr>`. Pro přehlednost je vhodné řádku ukončit značkou `</tr>`, nicméně není to nutné. V této značce můžeme definovat například barvu pozadí, zarovnání textu a obrázků v buňce (vlevo vpravo, uprostřed), vertikální zarovnání (uprostřed, nahoru, dolů) a další. Doposud ovšem prohlížeč nic nezobrazuje. Teprve po použití značky `<td>` bude text psaný po této značce (nebo jakýkoli jiný element HTML dokumentu) umístěn do buňky tabulky. Všeobecně platí, že by počet buněk měl být stejný na všech řádcích. Jestliže však chceme formátovat složitější tabulky, můžeme použít parametry značky `<td>` – `rowspan` a `colspan`. Tyto parametry nám definují, kolik buněk má aktuální buňka přesahovat a to ve směru vertikálním (`rowspan`) nebo horizontálním (`colspan`). Musíme mít na paměti, že značka obsahující např. `<td colspan=2>` funguje jako dvě značky `<td>`. V případě, že jsme na řádce měli tři buňky, budeme teď muset mít na řádce pouze dvě buňky (`<td colspan=2>` a `<td>`). To samé platí pro `rowspan`. Pro názornost uvedu příklad: Chceme vytvořit následující tabulku:

Zdrojový kód by pak vypadal následovně:

```
<table border="0" width="600"><!--tabulka nebude mít  
okraj a bude široká 600 pixelů-->  
  <tr>  
    <td>první buňka prvního řádku</td>  
    <td rowspan="2">druhá buňka přecházející do  
druhého řádku</td>  
    <td>třetí buňka</td>  
  </tr>  
  <tr>  
    <td>první buňka druhého řádku</td>
```

```
<td rowspan="2">ačkoli na druhém místě, nyní  
jsme ve třetí buňce - druhou tvoří buňka z předchozího  
řádku. Tato buňka bude zasahovat do dalšího řádku</td>  
</tr>  
<tr>  
<td colspan="2">buňka na prvním a druhém  
místě. Třetí buňka je tvořena převislou buňkou  
z minulého řádku </td></tr></table>
```

5.4.2 Program Převodník rozvrhů

Program vychází z několika předpokladů formátu vstupního souboru:

- existují dva typy tabulek. První je tabulka rozvrhů učeben:
 - Tabulka začíná řádkem, který začíná třemi znaky „+“ následovanými rozdělovačem „|“ a pak sedmi skupinami deseti znaků „+“ oddělovanými znaky „|“.
 - druhý řádek začíná názvem učebny, na něj navazuje bez oddělovače hodina prvního bloku (7:00) a po desetiznakových odstupech jsou hodiny oddělovány opět znakem „|“.
 - třetí řádek je totožný s prvním
 - 4., 7., 10., 13. a 16. řádek obsahuje nejprve zkratku dne (po, ut ...), oddělovač „:“ a pak vlastní kód předmětu. Oddělovačem mezi bloky je „:“
 - V případě že se v učebně střídá podle sudých a lichých týdnů jiná výuka, je další předmět uveden ve stejném formátu jako 4,7,10,13,16 (až na úvodní kód dne) i na 5., 8., 11., 14., 17. řádku.
 - 6., 9., 12., 15. a 18. Řádek obsahuje po třech mezerách znak „|“ a desetiznakové řetězce znaků „-“.
- Druhý typ tabulky - rozvrh studijních skupin je obdobný, pouze úvodní sloupec je široký 5 znaků a šířka dalších sloupců je jen 9 znaků.

Program potřebuje zadat jak vstupní adresář rozvrhů, tak výstupní adresář, kam může generovat HTML soubory. Na unixovském serveru OTUS, kde jsou stránky HF TU umístěny, již existuje emulátor Netware serveru, takže je možné specifikovat výstupní cestu přímo na tento server a ulehčit si od jinak nutného „uploadu“ (nahrání na server) pomocí FTP klienta. Program transformuje všechny rozvrhy (včetně jiných fakult), ale z WWW HF TU jsou aktivní pouze odkazy na studijní skupiny HF a rozvrhy učeben.

Jak program pracuje ? Nejprve zkontroluje, jestli byl správně spuštěn se dvěma parametry. V případě že ne, vypíše správnou syntaxi programu. V opačném případě začne načítat soubory ze zadaného vstupního adresáře. U souborů začínajících znaky UCEB (mají rozdílný formát tabulek než rozvrhy studijních skupin, viz výše) a nastaví proměnnou, ve které se nachází šířka sloupce na 10 a šířku prvního sloupce na 3 znaky. U studijních skupin je to 9 resp. 5 znaků. Zavolá proceduru na generování hlavičky HTML včetně jména generovaného podle názvu souboru. Pak zavolá proceduru *jeden_soubor* a poté pokračuje na další soubor. Na konci je generováno ukončení HTML dokumentu.

Procedura *jeden_soubor* kontroluje nastavení souborů, načítá vstupní soubor po řádce a spouští proceduru výměny jednoho řádku. V této proceduře dochází k vlastní záměně textu za text s HTML značkami. Procedura kontroluje, jestli se nachází někde v tabulce, nebo mezi tabulkami. Pakliže je mimo tabulky, ignoruje veškerý text který zde je (V případě že nejsou prvními znaky +++ , což znamená že začíná nová tabulka a do výstupního souboru se zapíše „<table ...>“. Toto testování se nachází až na konci procedury). V adresáři se totiž mohou nacházet i jiné soubory než tabulky s textem, ale vzhledem k možnosti zadání nových kruhů apod. jsem nechtěl, aby bylo nutné program předělávat v případě, že by bylo nutné přidávat další soubor ke konvertování (v současné době konvertuje vše, co je v adresáři; v případě že soubor neobsahuje tabulky, bude výsledek prázdný soubor). Když je v tabulce, nejprve přičte k počítadlu čísla řádku v tabulce číslo 1. (Existuje totiž skupina řádků se specifickými znaky, definujícími konec buňky

tabulky - viz formát tabulky v kapitole 5.4.2 - Program Převodník rozvrhů, strana 27). Následuje výběr akce pomocí funkce select case:

- když se nacházíme na 18. řádku tabulky, znamená to konec tabulky a proto se zapíše do výstupního souboru řádek s „</tr></table>“
- když jsme na 6., 9., 12. nebo 15. řádku, znamená to konec řádku v tabulce (a začátek dalšího). Tedy: „</tr><tr valign=top align=center>“
- na druhém řádku jsou informace o názvu kruhu, nebo učebny a časová pásma vyučovacích bloků.
- Jelikož jsou informace o jednom bloku ve dvou řádcích, (někde je druhý prázdný) je nutné, aby se první řádek nejprve zapsal do paměti, a teprve po přečtení dalšího řádku je možné vygenerovat HTML a zapsat ho jako řádek do výstupního souboru (viz formátování tabulek - kapitola 5.4.1, strana 25). První řádek jednoho dne (řádky tabulky 4., 7., 10., 13. a 16.) je tedy rozkouskován na buňky a po částech zapsán do pole.
- teprve při čtení řádků 5., 8., 11., 14. a 17. Je volána procedura rozvrh_dne která generuje celý řádek, včetně informací schovaných v poli.

Procedura rozvrh_dne nejprve vygeneruje začátek řádky - název dne pomocí rozhodování select case. Dokud není na konci řádku, zapisuje jednotlivé buňky (<td>) tabulky s daty z pole a druhého řádku, ze kterého je získává pomocí funkce copy.

5.5 Nástěnka HF

5.5.1 CGI

Common Gateway Interface neboli „všeobecné rozhraní pro brány“ je rozhraní umožňující spouštění programů na straně severu WWW klientem. Pro vlastní program pak platí, že musí umět brát informace tak, jak to definuje standard CGI a výsledkem jeho činnosti musí být výstup pochopitelný pro WWW browser. V 99% případů se bude jednat o HTML dokument. Program může být vytvořen v libovolném programovacím jazyce. V podstatě jedinou podmínkou je, aby byl jako takový spustitelný na serveru. (Například je možné aby byl vytvořen ve Visual Basicu, pak ale musí být server založen na platformě Windows a ne třeba Unixu, kde by se spustit VB program podařilo skutečně málokomu) Základní možností, která je často využívána je použití vlastního interpretu příkazů systému. Batch soubory DOSU jsou sice velmi ojedinělým příkladem, nicméně na platformě UNIXU sám interpret příkazů má dostatečné možnosti pro zpracování vstupů z HTML formuláře. Proces spuštění CGI skriptu má následující kroky:

1. Klient zašle serveru dotaz, jehož URL ukazuje na některý z CGI skriptů. Kromě toho může klient předat i doplňující data, ovlivňující činnost skriptu. (většinou je k zadávání těchto dat vytvořen HTML formulář, nebo je (třeba v případě sensitivního obrázku) generován samotným prohlížečem). Podmínkou však je, že každým výstupem CGI skriptu musí být objasnění obsahu výstupu programu. To je zajištěno první řádkou výstupu, ve kterém je uvedeno např. pro standardní HTML text následující:

```
Content-Type: text/html
```

(za prvním řádkem následuje jeden prázdný pro oddělení vlastní náplně výstupu od hlavičky)

2. Server zjistí, že požadované URL je třeba obsloužit externím programem. Spustí tedy dotyčný program a prostřednictvím CGI rozhraní mu předá informace, které bude potřebovat ke své činnosti.
3. CGI skript proběhne, zpracuje data a na jejich základě vytvoří HTML dokument. Ten odevzdá prostřednictvím svého standardního výstupu jako výsledek.
4. Server zašle klientovi odpověď, ve které mu předá výsledek CGI skriptu. Tím celá operace končí.

Z výše uvedených informací je zřejmé že toho při transakci dělá server více než při obvyklém dotazu na HTML dokument. Na serveru musí být nakonfigurováno, jak má poznat, že dokument na který je vznesen dotaz je CGI skript. Zároveň zde server **spouští program**, což samo o sobě přináší určitá bezpečnostní rizika.

Jelikož většina CGI skriptů zpracovává určitým způsobem data od klienta, je nutné zajistit předání těchto dat vlastnímu programu. Data posílaná CGI skriptu jsou kódována na základě pravidel používaných v URL. Všechny „zvláštní“ znaky jsou převedeny do dvojice číslic šestnáctkové číselné soustavy, před kterými je znak „%“. Existují dva typy požadavku klienta:

- metoda **GET** - při použití této metody jsou data, která mají ovlivnit činnost klienta, předána jako součást URL. Za cestu identifikující CGI skript, je připojen otazník a za něj jsou umístěna data. Mají-li několik položek, jsou tyto položky navzájem odděleny znakem „+“. Server analyzuje URL, spustí patřičný skript a data mu předá jako parametry na příkazovém řádku. Metoda GET je jednodušší, ale objem dat, který lze předat tímto způsobem, je omezen.
- metoda **POST** - Informace se předávají na standardní vstup CGI skriptu. Klient naváže spojení se serverem, zadá příkaz POST a poté odešle data



tímto spojením serveru. Ten na základě URL spustí CGI skript a data co přišla mu předá jako standardní vstup.

V obou případech server vytvoří celou řadu proměnných prostředí, která předá spouštěnému programu. Podle definice CGI verze 1.1 je server povinen nabídnout následující proměnné:

REQUEST_METHOD	Metoda předání dat; buď POST nebo GET
QUERY_STRING	data předaná v URL;
PATH_INFO	cesta, která má být zpracována skriptem;
PATH_TRANSLATED	cesta k souboru (v rámci systému souborů serveru), který má skript zpracovat; jedná se o hodnotu proměnné PATH_INFO, na kterou byly provedeny mapovací příkazy podle konfigurace serveru.
CONTENT_TYPE	MIME typ zasílaných dat (pro metodu POST)
CONTENT_LENGTH	délka zasílaných dat (POST)
SCRIPT_NAME	URL identifikující CGI skript
SERVER_NAME	jméno serveru
SERVER_PORT	číslo portu
SERVER_SOFTWARE	jméno a verze programu, realizujícího server
SERVER_PROTOCOL	jméno a verze protokolu; většinou se jedná o HTTP/1.0
GATEWAY_INTERFACE	jméno verze CGI, použitého pro spuštění skriptu; zpravidla CGI/1.1

REMOTE_HOST	doménové jméno počítače, ze kterého přišel dotaz
REMOTE_ADDR	IP adresa počítače, ze kterého přišel dotaz
AUTH_TYPE	způsob prokázání totožnosti
REMOTE_USER	má-li předchozí proměnná hodnotu „Basic“, obsahuje REMOTE_USER jméno uživatele tak, jak jej oznámil klient
REMOTE_IDENT	informace o identitě, získané dotazem podle RFC 931; server nemusí tuto proměnnou podporovat

5.5.2 (Ne)Bezpečnost CGI skriptů

Využití CGI skriptů je velmi rozsáhlé. Většinou se však používají v kombinaci se vstupem od uživatele. Vnáší tak příjemný prvek interaktivity do jinak statických WWW stránek. To s sebou však přináší i problémy bezpečnostního charakteru. Právě CGI skripty, které zpracovávají uživatelův vstup, jsou často terčem útoků hackerů. Hacker předpokládá, jak program pravděpodobně funguje a snaží se mu vsunout takový vstup, který mu umožní dělat věci ke kterým skript nebyl vytvořen. V naprosté většině případů se při této metodě využívá schopností většiny interpretů příkazové řádky zadávat více příkazů najednou. Tyto příkazy jsou oddělovány speciálními znaky a právě připojením tohoto znaku a příkazu připojeného hackerem může dojít k získání hodnotných informací - třeba o struktuře adresářů a souborů na serveru. Z předchozí části je zřejmé, že nejsnazším terčem jsou právě CGI skripty psané přímo v interpretu příkazového řádku či jiných nekompilovaných skriptů. Interpret lze však spustit i z kompilovaných CGI skriptů. Je proto minimálně nutné si ve svých skriptech na takovéto speciální znaky dát pozor. Buď v případě nalezení takovýchto znaků vůbec



skript nespouštět, nebo znak odstranit a pokračovat se vstupem, ve kterém jsou všechny nebezpečné znaky odstraněny.

5.5.3 Perl

Perl, neboli „*Practical Extraction and Report Language*“ (volně přeloženo jako „jazyk určený pro vytváření výtahů a zpráv“) je jak z názvu vyplývá programovací jazyk určený zejména pro práci s textem. [5] Právě proto je velmi vhodný právě pro zpracování CGI skriptů. Ačkoli jsem se zdráhal pouštět se do učení dalšího jazyka, nakonec jsem rád, že jsem si čas našel, neboť se jedná o velmi vstřícný jazyk, pomocí kterého je snadné vytvořit většinu toho, co jsem doposud dělal pomocí jiných prostředků, a to s daleko menší námahou. Já osobně pokládám za největší klad Perlu to, že umožňuje zapomenout na jakékoli hranice. Není nutné si dávat pozor aby hodnota proměnné nepřekonal povolenou mez. Většinou Perl žádné meze neklade. Jedinou bariérou je vlastní operační paměť systému.

Mluvím-li právě o systému, je třeba podotknout že programy Perlu jsou kompilovány až při spouštění programu. Vlastní „*compiler*“ je k dispozici pro nejrozličnější operační systémy, takže bylo možné, abych skripty testoval doma na systému Windows95 a po malých úpravách mohl být zprovozněn na serveru OTUS.

5.5.4 Skript

CGI skript, který obstarává chod nástěnky na WWW stránkách HF TU je řešen pomocí skriptového jazyka Perl. Ačkoli jsem původně chtěl vytvořit velice jednoduchý skript pomocí interpretu příkazů operačního systému UNIX, nakonec jsem od tohoto záměru upustil, neboť takto nelze ošetřit případ, že by se v jednom momentě snažilo do nástěnky poslat zprávu více uživatelů. Jestliže by toto nastalo, došlo by v lepším případě k znehodnocení jedné z posílaných zpráv, v horším pak ke smazání celého souboru se zprávami. Jelikož jsem nemohl použít můj oblíbený Visual Basic na platformě UNIX, musel jsem sáhnout po nástroji, který pro mne sice byl neznámý, na druhé straně byl ale všeobecně uznávaný jako velmi silný

nástroj právě v práci s řetězci a podobně, tedy ideální náhražka již zmíněného interpretu příkazů. Skript používá externích procedur ze souboru *cgi-lib.pl* pro zpracování QUERY_STRING (viz kapitola 5.5.1 - CGI, strana 30) a usnadňuje tak dramaticky práci s proměnnými.

Nejprve program vypíše standardní hlavičku každého CGI (`Content-type: text/html`). Pak následuje vlastní hlavička HTML dokumentu s názvem dokumentu atd. Zároveň je nutné vložit informaci o kódování češtiny. Ačkoli je skript v Perlu a běží na Linuxovském serveru, psal jsem ho pod operačním systémem Windows, takže je česká diakritika kódována na základě kódové stránky 1250. Snažil jsem se zachovat vzhled stránky jako celku, takže i zde je typické odsazení textu kromě nadpisů.

Celou nástěnku tvoří dva HTML soubory a jeden CGI skript. Jeden HTML dokument je formulář, kam jsou uživatelem vyplněna pole jméno, e-mail adresa a vlastní text. Uživatel musí vyplnit všechna pole. V případě, že je jedno z nich prázdné, oznámí mu to skript chybovým hlášením a odkazem zpět na formulář. Zároveň je pole textu kontrolováno na znaky „<“ a „>“ které by byly WWW prohlížečem chápány jako klasické HTML značky a bylo by možné do textu vkládat obrázky, formátovat libovolně text apod., což v případě veřejné nástěnky nechceme. V případě že je skript nalezne, opět toto oznámí chybovým hlášením.

Když je vše v pořádku, zjistí si datum platné na serveru, vytvoří kopii druhého HTML souboru, (na tomto místě probíhá testování, jestli v tu chvíli nečiní totéž jiný uživatel - v případě že ano, skript chvíli počká a pokusí se znova. Po třech nezdařených pokusech oznámí, že někdo jiný se pokouší zapisovat) ve kterém jsou vlastní zprávy nástěnky. V případě, že je vstup uživatele korektní, přidá za značku umístěnou v HTML (jedná se o komentář, takže je při formátování WWW prohlížečem ignorován) jméno autora, s odkazem na jeho adresu, datum a vlastní tělo zprávy.



5.5.5 Alternativní řešení

Pochopitelně není CGI jediným řešením zpracování formulářů. Jedním z možných je použití vsuvek serveru Apache, na kterém jsou WWW stránky HF TU umístěny. V současné době doplňuje několik základních vsuvek ještě nadstavba serveru Apache s názvem PHP/FI (další informace na <http://www.vex.net/php>), který navíc spolupracuje s databázovým systémem Mini SQL (<http://www.hughes.com.au>).

Výhoda spočívá v jejich přehledném umístění přímo ve zdroji HTML. Nevýhodou jsou podobně jako u CGI jistá bezpečnostní rizika a závislost na schopnostech serveru je zpracovat. Veškeré vsuvky jsou však umístěny ve standardním HTML okomentováním, takže v případě, že by jim server nerozuměl, jsou ignorovány. Podobných nadstaveb existuje více, záleží jen, na jakém serveru jsou stránky umístěny.

5.6 Vyhledávací služby

Aby uživatelé WWW stránek HF TU nemuseli při vyhledávání opouštět server OTUS, nabízí jim WWW HF TU seznam nejpoužívanějších vyhledávačů světa. Uživatel zadává hledaný řetězec do formuláře na domácím serveru, po odeslání však dotaz putuje na daný vyhledávač a ze vzdáleného serveru je pak uživateli předložen výsledek hledání.

5.6.1 PH Server

PH server slouží k vyhledávání informací týkajících se osob, majících účet na určitém serveru. Na Technické Universitě již tento server nějakou dobu běží a uživatelé mají možnost pomocí klienta získat informace o e-mailu, plném jménu, telefonním čísle a event. čísle kanceláře. Jelikož se jedná o software typu klient-server, je možné použít pro vyhledávání informací vlastního klienta. Opět zde fungují pravidla platformové nezávislosti. O možnosti vyhledávat tento typ informací na akademické půdě ví jenom hrstka lidí, takže bylo vcelku rozumné tuto



službu zpřístupnit většímu počtu event. zájemců. Existuje sice klient pro protokol Gopher, který bylo možné využít (v první fázi vývoje WWW stránek tomu tak opravdu bylo) pomocí odkazování se na gopherovské služby, nicméně například prohlížeč MS Internet Explorer nepodporuje tzv. CSO vyhledávání. Bylo nutné vytvořit buď celého klienta, nebo alespoň vytvořit CGI skript využívající klienta pro operační systém Unix. Protože jsem byl v časové tísní, rozhodl jsem se pro snadnější druhou variantu. CGI skript pouze předává uživatelský vstup klientovi PH a vrací nazpět výstup tohoto klienta. Kvůli bezpečnosti (viz kapitola 5.5.2 - (Ne)Bezpečnost CGI skriptů, strana 33) bylo nutné ještě přidat kontrolu vstupu uživatele. Opět kvůli nedostatku času je vlastní tělo odpovědi formátováno pomocí značky „<pre>“, tedy bez principů HTML dokumentu. Značka „<pre>“ totiž definuje, že jakýkoli text za touto značkou bude formátován jako obyčejný text. Do budoucna plánuji úpravu do vizuálně přitažlivější formy.

5.7 Další možnosti Intranetu

V názvu této kapitoly uvádím termín Intranet. Nejedná se o překlep. Intranet je termín označující lokální síť založenou na protokolu a službách Internetu. Tato lokální síť sice nemusí být k vlastnímu Internetu připojena, ale eventuelní připojení pak stojí minimum námahy. Navíc je zaručena platformová nezávislost, takže je možné v organizaci využívat najednou serverů s operačním systémem Unix, Windows NT, a dalšími najednou.

Univerzitní síť Liane je sítí lokální, ale několika přípojnými body je pevnou linkou připojena na uzly společnosti Cesnet, která patří mezi největší tuzemské poskytovatele Internetu. Vytvoření informačního systému založeného na TCP/IP je více než logické. Mnoho funkcí existujícího systému není prozatím WWW prezentací HF nahrazeno, proto bych se rád dalšímu rozvoji stránek věnoval během dalšího studia. Pochopitelně by bylo taktéž vhodné doplnit spektrum nabízených služeb o zcela nové. Velice užitečným by bylo vytvoření databáze absolventů hospodářské fakulty Technické University. Podobné indexy jsou ve



světě běžné a napomáhají zaměstnavatelům zjistit podrobnější informace o studentovi a tím pádem zároveň i uchazeči o zaměstnání. Ještě větším přínosem by byla „burza“ nabídek a poptávek po studentech na praxi v podnicích. Obor Podnikatelská Informatika má do osnov studia zařazenou dokonce dvousemestrovou řízenou praxi a podobný index by určitě napomohl při hledání vhodného podniku. Za těmito indexy by se skrývala databáze a uživatel by měl možnost vyhledávání záznamů podle různých kritérií. Na jednodušší databáze by bohatě stačil produkt MiniSQL, který je již na serveru OTUS zprovozněn.



6 Závěr

Cílem mého snažení bylo vytvoření WWW stránek s velkou dynamickou částí a převedení hlavních komunikačních služeb využívaných převážně díky lokální síti TU. Převedením na WWW a sloučením do jednoho prostředí získají tyto služby na užitečnosti a počítám i na intenzitě využívání. Velikou výhodou vidím taktéž ve zpřístupnění informací studentům středních škol a snad snížením vytížení studijního oddělení hospodářské fakulty. Velikou výhodou pro studenty HF jsou zase materiály týkající se jednotlivých vyučovaných předmětů.

Během tvorby WWW stránek jsem se seznámil s dvěma programovacími jazyky, nicméně daleko náročnější byly fáze rutinního formátování textu, kterých bylo bohužel mnoho.



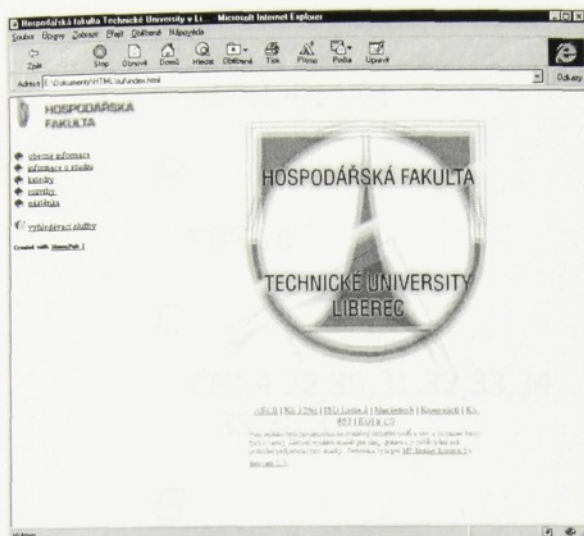
7 Literatura

- [1] Berners-Lee, T. - Connolly, D.: **Hypertext Markup Language - 2.0**. RFC 1866, 1995
<ftp://ftp.vse.cz/pub/docs/rfc/rfc1866.txt>
- [2] Berners-Lee, T. - Masinter, L. - McCahill, M.: **Uniform Resource Locators (URL)**.
RFC 1738, 1994, <ftp://ftp.vse.cz/pub/docs/rfc/rfc1738.txt>
- [3] Mansfield, R.: **The visual guide to Visual Basic**. Ventana Press, 1996,
ISBN 1-56604-192-9
- [4] Satrapa, P.: **World Wide Web - pro čtenáře, autory a misionáře**.
1996
- [5] Schwartz, R.: **Learning Perl**. O'Reilly & Associates, 1993
ISBN 1-56592-042-2

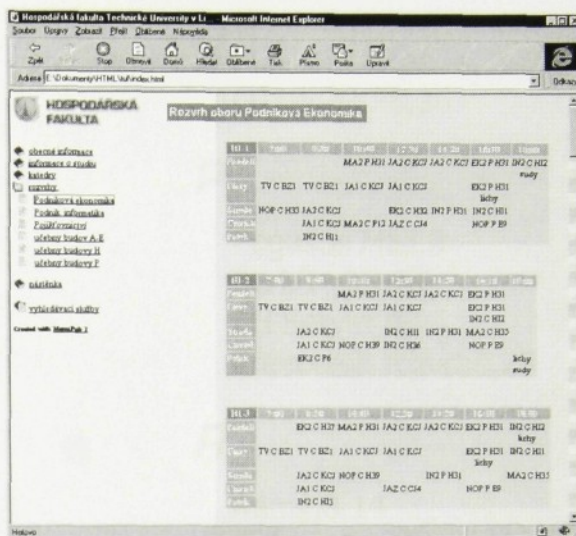


8 Přílohy:

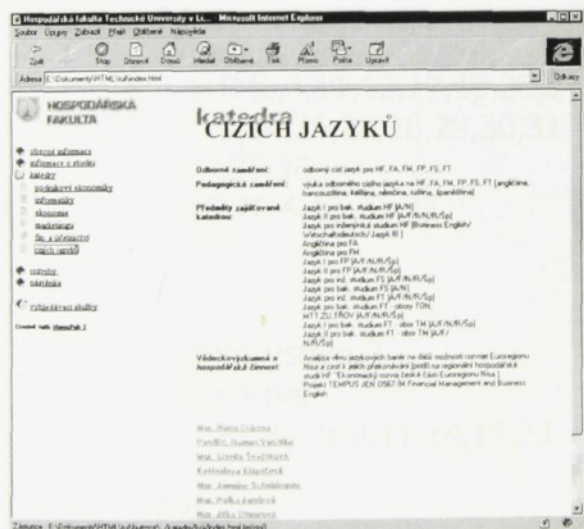
Ukázky WWW stránek HF



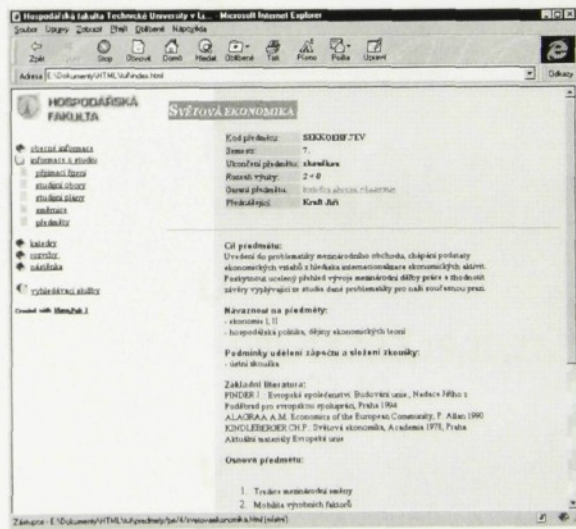
Úvodní obrazovka WWW stránek HF TU



Rozvrh oboru Podniková Ekonomika



Prezentace kateder



Jeden ze sylabů předmětů vyučovaných na HF



Rejstřík

A

ASCII,4,11,21

B

BBS,10

C

CGI,4,22,30,31,32,33,34,
35,36,37

G

Gopher,4,12,37

H

HTML,4,11,12,17,19,20,2
1,24,25,26,28,29,30,31
,35,36
HTTP,4,12,21,32

I

Infosystém,9
Internet
Explorer,8,11,14,15,23

M

MenuPub,20,21
Mini SQL,36

N

Netscape
Navigator,8,14,23,26

O

OTUS,22,28,36

P

Pegasus Mail,9
Perl,34,35,40
PH Server,40
Převodník rozvrhů,27

R

Rámy,8,14
Rozvrh,24,25

S

SaCzech,21,22

T

TYTO,23,24

U

URL,4,12,13,19,20,22,30,
31,32,40

V

Visual
Basic,18,25,30,34,40
Vyhledávací služby,36

W

WWW site,10,12,23