

**ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ**

KATEDRA APLIKOVANÉ ELEKTRONIKY A TELEKOMUNIKACÍ

BAKALÁŘSKÁ PRÁCE

vedoucí práce:

Doc. Ing. Jiří Masopust, CSc. 2018

autor:

David Forejtek

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta elektrotechnická

Akademický rok: 2017/2018

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **David FOREJTEK**

Osobní číslo: **E13B0182P**

Studijní program: **B2612 Elektrotechnika a informatika**

Studijní obor: **Elektronika a telekomunikace**

Název tématu: **IPTV vysílání FEL ZČU v Plzni**

Zadávací katedra: **Katedra aplikované elektroniky a telekomunikací**

Z á s a d y p r o v y p r a c o v á n í :

1. Prostudujte a popište možnosti šíření vlastního TV vysílání prostřednictvím internetu.
2. Zvolte adekvátní technologii pro podmínky FEL ZČU v Plzni a navrhnete potřebné komponenty pro realizaci.
3. Navrhnete programové schéma a začlenění do konceptu TV ZČU v Plzni.
4. Dle možností realizujte pilotní projekt vysílání.
5. Zhodnoťte dosažené výsledky.

Rozsah grafických prací: **podle doporučení vedoucího**

Rozsah kvalifikační práce: **30 - 40 stran**

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

Student si vhodnou literaturu vyhledá v dostupných pramenech podle doporučení vedoucího práce.

Vedoucí bakalářské práce: **Doc. Ing. Jiří Masopust, CSc.**

Katedra aplikované elektroniky a telekomunikací

Datum zadání bakalářské práce: **10. října 2017**

Termín odevzdání bakalářské práce: **7. června 2018**


Doc. Ing. Jiří Hammerbauer, Ph.D.
děkan




Doc. Dr. Ing. Vjačeslav Georgiev
vedoucí katedry

V Plzni dne 10. října 2017

Abstrakt

Předkládaná bakalářská práce je zaměřena na návrh a technickou realizaci internetové televize v síti FEL ZČU v Plzni.

Klíčová slova

Stream, IPTv, kodek, server, port, protokol

Abstract

The presented bachelor thesis is focused on the design and technical realization of Internet TV in FEL ZČU network in Pilsen.

Key words

Stream, IPTv, codec, server, port, protocol

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této bakalářské práce.

Dále prohlašuji, že veškerý software, použitý při řešení této bakalářské práce, je legální.

.....
podpis

V Plzni dne 11.6.2018

David Forejtek

Obsah

Originál (kopie) zadání BP/DP.....	2
Abstrakt.....	4
Prohlášení.....	6
Obsah.....	7
Úvod.....	9
1. Pojmy.....	11
1.1. Televizní vysílání (televize).....	11
1.2. Programové schéma.....	11
1.3. Server.....	12
1.4. Multimediální stream.....	12
1.5. Protokol.....	13
1.6. Multimediální kontejner.....	13
1.7. Kodek.....	13
2. Možnosti šíření TV signálu internetovou sítí.....	13
2.1. Šíření TV signálu internetovou sítí.....	13
2.2. Multimediální streaming internetovou sítí.....	15
3. Prvotní představa.....	15
4. Stavební kameny.....	16
4.1. LAMP.....	17
4.1.1 Linux.....	17
4.1.2 Apache.....	17
4.1.3 MySQL.....	17
4.1.4 PHP.....	17
4.2. Software pro streaming videa.....	17
4.2.1 Plex.....	17
4.2.2 Ffmpeg.....	18
4.2.3 Wowza.....	18
4.2.4 VLC.....	18
5. Funkční celky.....	18
5.1. Databázový systém.....	19
5.1.1 Struktura tabulky nahraných souborů.....	19
5.1.2 Struktura tabulky osob s vyšším stupněm oprávnění.....	20
5.1.3 Komunikace s databází.....	20
5.2. Skripty.....	21
5.2.1 Konfigurace php.ini.....	21
5.2.2 Upload.....	22
5.2.3 Sestavování vysílaného programu.....	25
5.3. Uživatelské rozhraní.....	28

5.3.1 Reportér.....	28
5.3.2 Redaktor.....	29
5.3.3 Divák.....	30
5.4. VLC.....	30
6. Programové schéma.....	30
6.1. Hrubé rozdělení.....	30
6.2. Hlavní vysílací čas.....	31
6.2.1 Dělení na základní vysílací formáty.....	31
6.2.2 Shrnutí.....	31
6.3. Tvorba denního programu.....	32
6.3.1 Předěly a jejich formáty.....	32
6.3.2 Denní informace.....	33
6.3.3 Shrnutí.....	33
7. Závěr.....	33
Seznam použitých zkratk.....	34
Seznam literatury a informačních zdrojů.....	35

Úvod

Potřebuje univerzita svoje televizní vysílání? Tato kapitola se v několika následujících odstavcích pokusí na tuto otázku alespoň nepřímo odpovědět tím, že objasní motivace a cíle projektu. Nejprve je ale třeba si alespoň ve stručnosti říci, co se televizním vysíláním míní a co nabízí.

Televize (internetová nebo „klasická“) s kontinuálním vysíláním tvoří významný informační zdroj, kde vedle obrazové a zvukové stopy lze zároveň přenášet i doprovodná data v nejrůznějších podobách, což přispívá k navýšení informačního obsahu média. Z tohoto hlediska se proto jedná o informačně robustní kanál, který si zaslouhuje svoji pozornost. Televizní vysílání a obecně televizi a její úkoly lze rozdělit následovně.

Vedle veřejnoprávní televize, jejímž hlavním cílem je mimo jiné „poskytování objektivních, ověřených, ve svém celku vyvážených a všestranných informací pro svobodné vytváření názorů“, [1] existují televize komerční, jejichž zájmem je především produkovat zisk poskytováním vysílacího času reklamním účelům. Komerční televize jsou v kontrastu k televizi veřejnoprávní soukromé. Mezi další podstatný rozdíl mezi těmito druhy televizí patří jejich financování; provoz veřejnoprávní televize financují koncesionářské poplatky, soukromé televize naproti tomu zisky plynoucí zejména z reklamy. Orgánem dohledu nad provozováním televizního vysílání je Rada pro rozhlasové a televizní vysílání. Rada mimo jiné uděluje, odnímá a mění licence k provozování vysílání a ukládá sankce a pokuty podle zákona. [2]

Podle způsobu vysílání lze vysílání rozdělit na analogové a digitální, přičemž analogové vysílání je v dnešní době vytlačeno. Mezi digitální vysílání patří systémy DVB-T, DVB-S, DVB-C a internetové televizní vysílání. [3]

Internetová televize k šíření vlastního vysílání využívá obecně počítačovou síť (internetovou, datovou). Počítačový server představující vysílací část řetězce v závislosti na použité technologii vysílání posílá síti za použití konkrétního internetového protokolu jednotlivé pakety obsahující multimediální datový tok určitého datového kontejneru. Tento datový tok může představovat na jedné straně buď živý stream nebo na straně druhé klientem vyžádaný proud dat. Podle způsobu šíření paketů sítí se pak dá vysílání rozdělit na unicast nebo multicast. Vysílání dále může být omezeno na konkrétní datovou síť v jednom případě a nebo v případě druhém, kdy je vysílání přístupné komukoli, hovoříme o webcastingu.

Cílem práce je technická realizace univerzitní internetové televize v síti FEL ZČU do které může každý student ZČU přispět vlastním autorským obsahem, nápadem na realizaci konkrétního projektu, popřípadě jinou tvůrčí činností, která povede k rozšíření nebo zkvalitnění programového schématu. Práce si neklade za cíl řešit organizační stránku projektu ani formální mechanismy volby vhodného obsahu.

Protože univerzita je instituce zákonem popsána jako: „nejvyšší článek vzdělávací soustavy jsou vrcholnými centry vzdělanosti, nezávislého poznání a tvůrčí činnosti a mají klíčovou úlohu ve vědeckém, kulturním, sociálním a ekonomickém rozvoji společnosti tím, že a) uchovávají a rozhojňují dosažené poznání a podle svého typu a zaměření pěstují činnost vědeckou, výzkumnou, vývojovou, uměleckou nebo další tvůrčí činnost“ [4]. Jako taková by se tedy univerzita mohla stát garantem mimo jiné relevantního a formálně seriózního obsahu vzdělávacího charakteru. Tolik k motivaci práce.

Vzhledem k výše popsanému může mezi možné přínosy práce patřit jednak zkvalitnění výuky vytvořením kvalitního multimediálního archivu, dále pak vytvoření mediálního prostoru pro například kulturní, sportovní nebo diskusní činnost studentů univerzity a v neposlední řadě reprezentace ZČU jako instituce.

1. Pojmy

Následující kapitola popíše význam důležitých pojmů užitých v práci. Pojmy související jednak se streamováním multimediálního obsahu internetovou sítí, dále pak s televizním vysíláním samotným a jeho strukturou.

Popsány budou následující termíny: televizní vysílání (televize), programové schéma, server, multimediální stream, protokol, multimediální kontejner a kodek.

1.1. Televizní vysílání (televize)

„Slovo televize pochází ze starořeckého τῆλε (tèle) - daleko a latinského visio - vidět. Televize patří mezi masová média (masmédia). Tuto funkci plní tvorbou a vysíláním televizního programu. Z hlediska sociální komunikace se jedná se o jednosměrné, dálkové, plošné vysílání (tzv. broadcasting), jehož prostřednictvím přijímá komunikant zvuk a obraz na svém přijímači. Příjem může být buď černobílý nebo barevný, dvou či trojrozměrný. Televize slouží především k zábavě, vzdělání, přenosu zpráv a reklamy. V užším pojetí může být televize chápána jako technický systém přenášející obrazy na dálku prostřednictvím televizního signálu, resp. televizního přístroje. V širších chápání je televize sociálním systémem masové komunikace, který zahrnuje technický systém, výsledný produkt (=vysílaný program), ale také organizační struktury, které jsou nutné k výrobě programu a jeho masové distribuci.“ [5]

K výše uvedené definici lze doplnit, že kromě obrazových a zvukových dat lze přenášet další doprovodná data jako například teletext nebo titulky pro neslyšící.

Internetová televize jako fenomén navíc rozšiřuje uvedenou definici v tom smyslu, že už se nemusí jednat pouze o jednosměrný informační kanál; konzument si může například pomocí technologie VOD vyžádat přehrání jen určitého obsahu, a celková možnost interakce konzumenta s platformou je vyšší než u klasického televizního vysílání. Dalším zásadním rozdílem mezi klasickým televizním vysíláním a IpTV je užití přenosové médium.

1.2. Programové schéma

„Programové schéma je princip zařazování televizních pořadů do určitých časových pásem v rámci jednotlivých dnů týdne. Televizní stanice si vytvářejí programová schémata, do nichž zařazují jednotlivé vysílané pořady. Programování vysílání nebo plánování, je

uspořádání televizních programů v denním, týdenním nebo sezónním programovém schématu.“ [strana 23 Vývoj programového schématu prime time TV Nova od jejího vzniku do současnosti Kateřina Michailidu]

1.3. Server

Server je počítač (nebo počítačový program) poskytující služby klientům na základě jejich požadavků; hovoříme o modelu klient-server. Tento model je možné rozdělit na: tenký klient a tlustý klient.

Tenký klient je označení pro případ, kdy zpracování požadavku spadá čistě do režie jiného počítače (serveru), kdežto u tlustého klienta není třeba dalšího počítače pro zpracování zadaného požadavku. Příkladem může být zpracování požadavku nad databázovým systémem. V případě, že klient pošle pouze požadavek a server vrátí výsledek, hovoříme o tenkém klientovi. Příkladem tenkého klienta je webový prohlížeč [<https://it-slovník.cz/pojem/tenky-klient>]. V případě, kdy by nejprve došlo ke stažení celé databáze na klientovu stranu a na ní by byl požadavek zpracován, hovořili bychom o modelu tlustý klient. Server v tomto případě slouží pouze centrální prvek (například jako sdílené úložiště). [6]

Služby klientům poskytuje program označovaný jako daemon (Unixové systémy), nebo jako služba (systémy Windows). Mezi nejznámější serverové demony se řadí Apache a Nginx. K poskytování služeb jsou využity internetové protokoly (HTTP pro zobrazování internetového obsahu, FTP pro přenos souborů, RTP pro přenos multimediálního obsahu v reálném čase a mnoho dalších). [7]

1.4. Multimediální stream

Jedná se o proud multimediálních dat posílaných počítačovou sítí. Slovem multimediální se míní, že data obsahují zvukovou, obrazovou a datovou složku. Datový tok je sítí posílán jako série paketů za pomoci určitého internetového protokolu. [8, 9]

Data mohou být přenášena různými způsoby; například lze přenášet obraz, zvuk a data separátně na různě očíslované porty, popřípadě lze obrazová a zvuková data přenášet zapouzdřena pomocí konkrétního kontejneru společně na jediný port. Další částí multimediálního streamu jsou metadata obsahující informace popisující soubor jako takový.

1.5. Protokol

Protokol definuje způsob komunikace mezi dvěma elektronickými zařízeními; stanovuje syntaxi, synchronizaci a podobně. Může být realizován na softwarové nebo hardwarové úrovni, popřípadě na kombinaci obojího. [10, 11]

1.6. Multimediální kontejner

Kontejnerem rozumíme soubor v kterém jsou pohromadě uložena obrazová, zvuková a popřípadě doprovodná data ve formě titulků a podobně. Další informace nesená kontejnerem je pak použitý kodek u každé stopy. Mezi známé formáty kontejnerů patří například: AVI, MPEG-4 part 14 (mp4), MPEG-PS, MPEG-TS, VOB a mnoho dalších.

1.7. Kodek

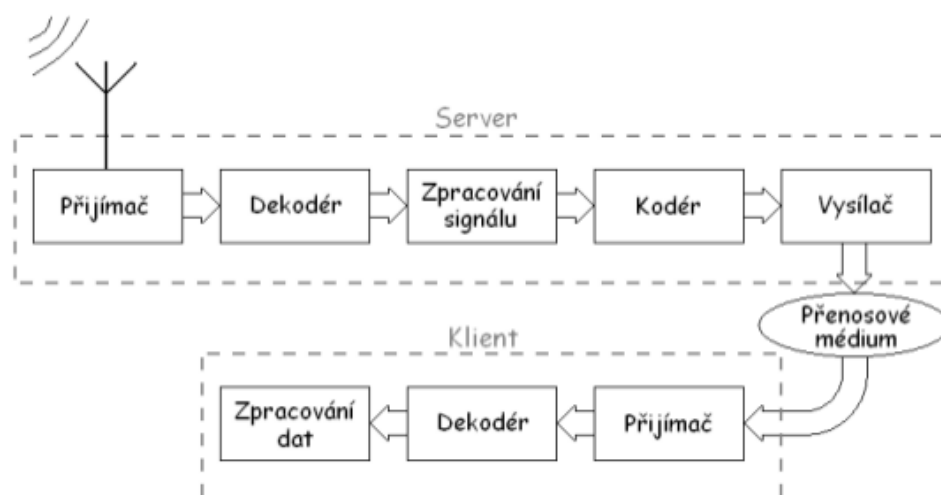
Slovo kodek vzniklo sloužením slov kodér a dekodér. Kodek představuje algoritmus (počítačový program) kterým jsou zvuková/obrazová data zapisována (enkódována) nebo čtena (dekódována). Kodek dále definuje vnitřní kompresi dat - z tohoto pohledu rozlišujeme bezztrátové a ztrátové kodeky. Mezi zvukové kodeky patří například: MP3 (ztrátový), AAC (ztrátový), Vorbis (ztrátový), Flac (bezeztrátový), AAC, MP2 (MPEG-2 Audio Layer II). Mezi obrazové kodeky pak: H.264, H.265, MPEG-4 Part II, XviD/Divx.

2. Možnosti šíření TV signálu internetovou sítí

Kapitola popíše jakým způsobem lze šířit vlastní televizní signál internetovou sítí a jaké funkční bloky jsou k přenosu potřeba a jak se přenos realizuje, přičemž pozornost bude zejména věnována multimediálnímu streamování. Tématu šíření vlastního TV signálu internetovou sítí a její realizaci se obsáhle věnuje diplomová práce TV vysílání v IP sítích Martina Kováře, a proto mu zde bude věnována jen okrajová pozornost.

2.1. Šíření TV signálu internetovou sítí

Protože televizní signál šířený technologií DVB-T, DVB-S nebo DVB-C se jako takový nehodí k šíření internetovou sítí a je nutné ho před přenosem nejprve zpracovat, je pro názornou představu vhodné uvést blokové schéma přenosového řetězce.



Obrázek 1: Přenosový řetězec (převzato z [Diplomové práce: „TV vysílání v IP sítích“, Martin Kovář])

1. Server

Po přijetí TV signálu a jeho převodu na datový tok vstupuje tento do bloku dekodéru. Blok dekodéru se stará o zapouzdření toku do paketů, při vyšším počtu kanálů o jejich výběr apod. Následně je možné s tímto tokem dále pracovat v bloku zpracování dat, informace v něm obsažené je možné převést do jiného kodeku a podobně. Pro přenos počítačovou sítí je následně v bloku kodéru zapotřebí tok rozdělit do paketů určitého protokolu, který je možné přenášet počítačovou sítí. Tento proud paketů pak vstupuje do síťového rozhraní odkud je poslán přenosovým médium sítí ke klientovi nebo klientům.

2. Přenosové médium

„Přenosové médium zde reprezentuje počítačová síť, založená většinou na technologii Ethernet a sady TCP/IP paketů. V těchto sítích se datový tok může šířit dvěma způsoby a to vysíláním skupinovým (multicast) nebo s jedním cílem (unicast)“ [citace z DP Martina Kováře]

3. Klient

Na straně klienta nejprve vstupuje tok paketů do přijímače (síťového rozhraní), odkud jednotlivé pakety pokračují do dekodéru, kde jsou dekodovány ochranné kódy. Dekodovaný signál může pokračovat do bloku zpracování dat. V této fázi jsou data připravena pro přehrávání nebo dalšímu zpracování.

2.2. Multimediální streaming internetovou sítí

V případě multimediálního streamingu je situace prakticky stejná jako v předchozím případě, odpadá ovšem příjem DVB signálu a jeho dekódování, protože v tomto případě se sítí posílají multimediální data uložená na serveru v konkrétním datovém kontejneru. Streamované soubory je opět nutné paketizovat a odesílat za použití vhodného protokolu sítí, o což se stará vhodný počítačový program označen blokem kodér. Stejně jako v příkladu výše se k šíření sítí může využít metody unicast nebo multicast, jednotlivé pakety jsou směrovány sítí ke konkrétním uživatelům nebo k určité skupině. Na straně klienta je zpracování identické s výše uvedenou metodou streamování TV signálu.

3. Prvotní představa

Cílem této kapitoly je popsat fungování internetové televize v síti FEL ZČU a v obecných rysech představit funkci dílčích komponenty které povedou k její realizaci.

Jednotlivá konkrétní řešení budou diskutována v k tomu určených kapitolách.

Jak již bylo řečeno, televize má uživatelům nabízet:

1. sledovat televizní vysílání
2. přispívat vlastním autorským obsahem

Pro splnění těchto požadavků je třeba navrhnout architekturu implementující následující funkcionality:

1. dynamické vytváření vysílaného programu na základě programového schématu
2. kontinuální streaming multimediálního datového toku ve vhodném formátu
3. upload multimediálního obsahu od uživatelů (studentů)
4. zanášení záznamů o těchto souborech do databázového systému
5. třídění a mazání uloženého obsahu
6. rozdělení oprávnění práce se soubory
7. uživatelské rozhraní pro skupiny s různou úrovní oprávnění (systém reportér/redaktor)

Dalšími požadavky na systém jsou:

1. bezobslužnost
2. jednoduchost
3. spolehlivost

Fungování systému popisuje následující sekvence procesů. Pořadí je stejné jako pořadí systémových funkcionalit výše:

1. z nahraného a schváleného multimediálního obsahu se pravidelně sestavuje vysílání. Požadavkem na tento proces je bezobslužnost; z informací o jednotlivých souborech uložených v databázovém systému se podle programového schématu a co nejjednodušší logiky vytváří vysílaný program.
2. Sestavený program jakožto posloupnost videí je následně streamován patřičným softwarem sítí FEL ZČU.
3. Uživatelé sítě mohou nahrát svůj autorský obsah na stranu serveru. Aby jim to bylo umožněno, musejí se přihlásit přes systém webAuth pomocí jednotného identifikačního univerzitního systému. Součástí procesu je kontrola parametrů nahrávaného souboru.
4. Při úspěšném nahrání se o souboru vytvoří záznam v databázovém systému. Záznam musí jednoznačně specifikovat nahraný soubor a obsahovat potřebné informace nutné pro případné zařazení do vysílání.
5. V součinnosti s databázovým systémem a na základě patřičných kritérií se pravidelně provádí uvolňování místa na disku. Soubory jsou přesouvány mezi adresáři podle toho, zda jsou nebo nejsou schváleny jako vysílatelné.
6. Po přihlášení popsaného v kroku 3, se rozděluje míra oprávnění práce se soubory na základě jmenného seznamu uživatelů s vyšším stupněm oprávnění. Tento seznam je uložen v databázovém systému.
7. Uživatelské rozhraní pro uživatele s oprávněním schvalovat vysílatelnost obsahu nabízí jiné možnosti než rozhraní pro uživatele oprávněné soubory pouze nahrávat.

4. Stavební kameny

V předchozí kapitole byla nastíněna obecná představa fungování IpTV v síti FEL ZČU. Cílem následující kapitoly je vybrat potřebné konkrétní softwarové nástroje.

Potřebám této práce plně vyhovuje kombinace softwaru označená zkratkou LAMP (Linux, Apache, MySQL, PHP) používající se například pro dynamické webové aplikace. Výhodou této sady softwarového vybavení je fakt, že se jedná o svobodný a volně šiřitelný software.

4.1. LAMP

4.1.1 Linux

Linux je nekomerční, svobodný a otevřený víceuživatelský a víceúlohový operační systém. Byl zvolen pro svou stabilitu, spolehlivost a z důvodů, že obsahuje potřebné podpůrné nástroje.

4.1.2 Apache

Apache je serverový daemon – program vyřizující požadavky klientů. Jedná se o zdaleka nejpoužívanější webový server používaný u asi 47% webových serverů [12].

4.1.3 MySQL

MySQL je relační databázový model využívající standardy SQL jazyka. „Základem relačních databází jsou databázové tabulky“ [13] mezi kterými existuje jistá relace. Tabulky jsou složeny z řádků představujících jednotlivé záznamy a sloupců, které představují jednotlivé vlastnosti záznamu. Každý sloupec je specifikován jedinečným názvem a datovým typem. MySQL je nejoblíbenějším databázovým systémem používaným velkými společnostmi jako je Facebook, Youtube a podobně [14].

4.1.4 PHP

PHP je programovací jazyk pracující na straně serveru, [15] který umožňuje vývoj dynamických webových aplikací. Další funkce kterou PHP nabízí je tzv. “Command line scripting“, kdy k spuštění skriptu není nutný ani webový prohlížeč ani server; skript běží lokálně na daném stroji. Takovéto skripty je možné spouštět nezávisle na serveru pomocí například plánovače úloh [16].

4.2. Software pro streaming videa

K tomuto softwarovému vybavení je třeba ještě přidat nástroj umožňující streaming videa. Existuje celá řada volně šiřitelných i komerčních řešení, patří mezi ně například:

4.2.1 Plex

Plex je volně stažitelný multiplatformní streamovací server určen pro sdílení multimediálních souborů v domácích sítích (s podporou transkódování před streamováním). Plex nabízí podporu pro celou řadu zařízení: od chytrých televizorů, přes

zařízení s operačním systémem Android, iOS až po PlayStation3 a 4 a celou řadu dalších. O serverovou část se stará Plex Media Server, o klientskou pak Plex Media Player nabízející pohodlné grafické rozhraní. Pro streamování Plex využívá protokoly NAT-PMP a UPnP. [17]

4.2.2 Ffmpeg

Ffmpeg je multiplatformní software šířený pod licencí LGPL, který umožňuje nahrávat, transkódovat a streamovat multimediální soubory. Program nedisponuje grafickým rozhraním. O rozšiřování funkcionalit, podporovaných kodeků a podobně se stará celá řada knihoven. [18]

4.2.3 Wowza

Wowza je multiplatformní placený streamovací software umožňující kromě živého streamování i VOD. Wowza podporuje pro streamování celou řadu protokolů (RTMP, RTSP/RTP, WebRTC, MPEG-DASH, HLS, HDS), před streamováním dokáže také transkódovat. [19]

4.2.4 VLC

VLC je multiplatformní volně šiřitelný open source software šířený pod licencí GPL. Kromě podpory přehrávání velkého množství formátů, umožňuje VLC podobně jako ffmpeg transkódovat a streamovat multimediální soubory. VLC také umožňuje otevření síťového proudu (streamu). Program disponuje grafickým rozhraním, ale lze ho ovládat i pomocí příkazové řádky.

Jako streamovací software byl nakonec zvolen VLC od neziskové společnosti VideoLAN a to protože je volně šiřitelný pod licencí GNU/GPLv2, splňuje potřebné funkce pro potřeby této práce, nabízí celou řadu dalších funkcí (transkódování, streaming ze zachytávacího zařízení...) potenciálně využitelných při budoucím rozšiřování projektu.

5. Funkční celky

Kapitola si klade za cíl popsat jakým způsobem budou použity jednotlivé softwarové nástroje, jakou funkci budou zastávat a jak budou propojeny do funkčních celků. Obecné představy nastíněné ve 4. kapitole zde budou diskutovány podrobněji.

5.1. Databázový systém

Pro potřeby této práce je nutné zamyslet se nad tím, kolik tabulek v databázovém systému je třeba vytvořit a jakou mají mít strukturu. Jak již bylo řečeno, databázový systém musí přinejmenším specifikovat nahrané soubory na serverovou stranu a osoby s vyšším stupněm oprávnění. V dalším kroku je nutné navrhnout vhodnou strukturu tabulek databáze.

5.1.1 Struktura tabulky nahraných souborů

K charakterizaci nahraných souborů je potřeba zaznamenat následující údaje v daných datových typech a v dané znakové sadě, která respektuje možnost použití diakritiky.

1. Vygenerované jedinečné jméno souboru – varchar(30) - utf8_general_ci
2. Původní jméno souboru – TINYTEXT – utf8_czech_ci
3. Kdo soubor nahrál – TINYTEXT – utf8_czech_ci
4. Kdy byl soubor nahrán – DATETIME
5. Délka videa – TIME
6. Schváleno/neschváleno – BOOL
7. Zařazení do programového schématu – varchar(30) – utf8_czech_ci
8. Komentář autora – TEXT – utf8_czech_ci
9. Komentář redaktora – TEXT – utf8_czech_ci

Záznamy v tabulce jsou následujících datových typů:

- BOOL – datový typ nabývající hodnot TRUE (nenulová hodnota), FALSE (nulová hodnota) nebo NULL. Všechny případy jsou v návrhu respektovány. TRUE značí schválené video, FALSE neschválené, NULL nezkontrolované. Datový typ bool zabírá jeden byte paměti a je ekvivalentní datovému typu TINYINT(1).
- DATETIME – datový typ ukládající záznamy ve formátu YYYY-MM-DD HH:MM:SS. Využívá se celkem 8 bytů, 4 bytový integer pro YYYY-MM-DD a 4 bytový integer pro HH:MM:SS.
- TIME – datový typ ukládající záznamy ve formátu HH:MM:SS. K uložení se využívá 3 bytový integer.
- TINYTEXT – tento datový typ je schopný uložit až 255 ($2^8 - 1$) znaků. Každý TINYTEXT má svůj 1 bytový prefix v němž je uložen počet nesených znaků. Každý znak je potom uložen pomocí jednoho bytu.

- TEXT – datový typ schopný uložit až 65535 ($2^{16} - 1$) znaků. Podobně jako TINYTEXT i TEXT má svůj prefix (tentokrát 2 bytový) nesoucí informaci o počtu uložených znaků. Každý znak je pak uložen pomocí jednoho bytu.
- varchar(30) – tento datový typ je schopen uložit až 30 znaků. Paměťová náročnost závisí na skutečném počtu uložených znaků. V případě, že je využita celá kapacita (v tomto případě 30 znaků), je potřeba 31 bytů paměti.

Architektura počítá s použitím dvou znakových sad:

- utf8_general_ci – mnohojazyčná znaková sada nerozlišující velikost písmen
- utf8_czech_ci – sada výhradně pro češtinu nerozlišující velikost písmen

Údaj o tom, kdo soubor nahrál se získá díky nutnosti být při nahrávání obsahu přihlášen systémem WebAuth. Komentář autora vyplní student při nahrávání obsahu, komentář redaktora bude do databáze zanesen při revizi obsahu osobou s vyšším stupněm oprávnění. Položka Schváleno/neschváleno je implicitně nastavena na NULL. Preferovanou vysílací dobu volí autor z výčtu možností (v závislosti na struktuře programového schématu) při nahrávání obsahu a redaktor má právo ji změnit.

Výše uvedené informace specifikují soubor dostatečně. Z pole zařazení do programového schématu, délky videa a Schváleno/neschváleno lze určit zda a v jakou dobu se má video vysílat.

5.1.2 Struktura tabulky osob s vyšším stupněm oprávnění

V případě struktury této tabulky je situace jednodušší než v případě předchozím. Pro specifikaci osob s vyšším stupněm oprávnění si vystačíme se jmenným seznamem těchto osob. Při přihlášení přes WebAuth potom systém zkontroluje, zda jméno přihlášené osoby odpovídá záznamu v tabulce a pokud ano, budou této osobě přidělena vyšší oprávnění a komunikace se systémem bude vedena přes odlišné rozhraní. Struktura je následující

1. orion_login - TINYTEXT

5.1.3 Komunikace s databází

Komunikaci s databází mají na starost jednotlivé PHP skripty. S databázovým systémem je třeba komunikovat v následujících případech:

1. Upload souboru

Záznam se v tabulce databáze vytvoří jen při úspěšném nahrání souboru na serverovou stranu a je v režii příslušného PHP skriptu.

2. Schválení souboru jako vysílatelného a revize obsahu

V případě, že redaktor schválí video jako vysílatelné, je nutné u tohoto videa změnit hodnotu v sloupci Schváleno/neschváleno z hodnoty FALSE na TRUE. Redaktor schvaluje nebo zamítá vysílatelnost pořadů pomocí uživatelského rozhraní. V případě revize obsahu má redaktor oprávnění měnit položky Preferovaná vysílací doba a Komentář redaktora

3. Sestavování vysílaného programu

V tomto případě se z databáze pouze čtou údaje u schválených videí o jejich délce a preferované vysílací době.

4. Smazání souboru

V případě smazání konkrétního souboru ze serverového úložiště je třeba smazat i příslušný záznam v databázovém systému.

V každém případě, kdy se z databáze vyčítají a nebo do databáze zapisují textová pole obsahující české znaky, je nutné do obslužného php skriptu vložit následující řádek kódu:

```
mysqli_set_charset($conn,"utf8")
```

ten zajistí nastavení znakové sady na utf8. \$conn je otevřené připojení k MySQL databázi.

5.2. Skripty

V dalším textu budou představeny jednotlivé skripty a bude uvedena jejich funkčnost. Text objasní nastavení konfiguračního souboru php.ini tak, aby byly splněny podmínky zejména pro možnost uploadu souboru na serverovou stranu, představí skripty pro upload, sestavování programu a uvolňování diskového prostoru.

5.2.1 Konfigurace php.ini

Úprava tohoto konfiguračního souboru slouží zejména k, jak už bylo řečeno výše, umožnění nahrávání souborů na server.

Aby byl upload souboru možný, je třeba v tomto konfiguračním souboru upravit klíčové direktivy a to zejména file_uploads a to na hodnotu On. Mezi další důležité direktivy patří ty omezující velikost nahrávaného souboru, jsou to: direktiva určující maximální velikost nahrávaného souboru upload_max_filesize, direktiva omezující velikost všech formulářových dat post_max_size a memory_limit určující maximální velikost paměti v bytech kterou skript může alokovat.

Mezi těmito direktivami by měl platit následující vztah: memory_limit > post_max_filesize > upload_max_filesize [19, 20].

Dalšími důležitými direktivami jsou `upload_tmp_dir` a `max_file_uploads`. První direktiva určuje cestu k dočasnému souboru pro uploadovaný soubor PHP skriptem a druhá maximální počet uploadovaných souborů pomocí jednoho požadavku.

5.2.2 Upload

PHP ve spolupráci s HTML formulářem umožňuje poměrně jednoduchým způsobem zajistit funkci nahrání souboru na server. Při potřebě zobrazit průběh uploadu souboru ve formě progress baru je dále nutné do kódu začlenit i část napsanou technologií AJAX. Skript dále musí komunikovat s databázovým systémem. Obecná představa o fungování skriptu je následující:

Uživatel přihlášený přes své Orion konto pomocí HTML formuláře vybere požadovaný soubor a potvrdí jeho nahrávání. Pokud soubor splňuje nutné náležitosti (požadavek na typ souboru a jeho maximální velikost) je přistoupeno k jeho uploadu. Pomocí technologie AJAX je průběžně zobrazován počet již přenesených bytů ve formě progress baru. Jestliže je soubor úspěšně přenesen, je o něm vytvořen záznam v tabulce databázového systému tak, jak bylo popsáno v kapitole 6.1.1 Konkrétní řešení bude uvedeno v dalším textu strukturovaném hierarchicky v pořadí HTML část, část JavaScriptu, PHP část.

1. HTML část

Pro představení funkce skriptu se nelze vyhnout popsání HTML formuláře, který je ke skriptu přidružen. HTML formulář má následující konkrétní tvar:

```
<form id="upload_form" enctype="multipart/form-data" method="post">
  <input type="file" name="file1" id="file1"><br>
  Komentář autora: <input type="text" name="koment_aura" id="koment_aura"><br>
  Zařazení programu: <select name="Zarazeni" id="Zarazeni">
    <option disabled selected="true" value="0"> -- Vyberte možnost -- </option>
    <option value="A">A</option>
    <option value="B">B</option>
    <option value="C">C</option>
    <option value="D">D</option>
  </select><br>
  <input type="button" id="button1" value="Upload File" onclick="uploadFile()">
  <progress id="progressBar" value="0" max="100" style="width:300px;"></progress>
  <h3 id="status"></h3>
```

```
<p id="loaded_n_total"></p>
</form>
```

Atribut ENCTYPE se při přenosu souborů nastavuje na "multipart/form-data", doporučená metoda přenosu je POST. Formulář dále specifikuje tři input tagy sloužící uživateli pro vkládání dat [21]. První, pojmenovaný file1, je typu file a je určený k manipulaci s nahrávaným souborem a je na něj odkazováno v přidruženém kódu napsaném v JavaScriptu. Druhý tag je typu button, vytváří tlačítko, po jehož stisknutí je volána funkce uploadFile(), která bude popsána dále. Třetí input tag je typu text a slouží uživateli k vložení textového komentáře. Dále formulář specifikuje tag select, kde z množiny přednastavených možností uživatel vybere jednu variantu popisující zařazení programu. Součástí formuláře je i progress bar, kterým bude zobrazen průběh uploadu.

2. Část v jazyce JavaScript

Konkrétní podoba kódu je uvedena v příloze, v tomto textu bude popsána funkčnost jednotlivých dílčích celků. Kód je napsaný pomocí technologie AJAX a je prakticky v nezměněné podobě přejat z [22]. Skript je doplněn o kontrolu typu a velikosti souboru a v době nahrávání souboru deaktivuje tlačítko button1. Pokud tlačítko zůstávalo aktivní, mohlo se stát, že při jeho opětovném zmáčknutí došlo v průběhu nahrávání k nahrávání dalšímu. Samotný kód je rozdělen na několik funkčních celků a to na funkce: uploadFile, progressHandler, completeHandler, abortHandler, errorHandler a pomocnou funkci pojmenovanou jako podtržítka (_).

Základem skriptu je funkce uploadFile(), která na začátku vytvoří proměnou file, do které jsou nahrány informace o uploadovaném souboru vyčtené z HTML formuláře na základě id příslušné položky file. V dalším kroku je zkontrolována velikost a MIME typ souboru. V případě příliš velkého souboru nebo souboru špatného typu je funkce v tuto chvíli ukončena a vypíše se chybová hláška, jinak funkce pokračuje dál. Následně je zneaktivněno tlačítko, vytvořen objekt formData typu FormData, do něhož jsou nahrány údaje o uploadovaném souboru pomocí metody append. Dále je vytvořen objekt ajax typu XMLHttpRequest, k němuž jsou přiřazeny jednotlivé EventListenery (progress, load, error, abort) metodou addEventListener. Objekt ajax následně otvírá metodou POST php skript starající se o samotný upload souboru. V konečném kroku jsou data uložená v objektu form poslána objektem ajax metodou send.

Součástí kódu jsou dále funkce obsluhující jednotlivé asynchronní události. Jsou to: progressHandler, completeHandler, errorHandler a abortHandler. Funkce progressHandler vypočítává procentuální průběh uploadu na základě informací o počtu přenesených bytů a

celkovém počtu bytů. Funkce dále upravuje atribut value progress baru, čímž dochází k jeho vykreslování. Funkce errorHandler a abortHandler jsou formálně shodné, při jejich volání dojde k vypsání textu informujícího o chybě nebo přerušení přenosu a zaktivní tlačítko umožňující zahájení nahrávání souboru. Funkce completeHandler při svém spuštění nastavuje hodnotu value progress baru na nulu a zaktivňuje uploadovací tlačítko.

3. PHP část

Součástí skriptu je kontrola, zda byl vůbec nějaký soubor pomocí HTML formuláře vybrán. Pokud ne, je vypsána chybová hláška a skript je ukončen funkcí exit. Samotný upload souboru je realizován funkcí move_upload_file. Parametry funkce jsou: dočasný adresář, kde je soubor nahrán a cílová cesta k souboru. Cílová cesta je složena z adresáře, kam se soubor nahraje a jeho jména, které je vygenerováno pomocí funkce unique.

V případě úspěšného nahrání je třeba provést dodatečnou kontrolu typu souboru, protože kontrola MIME typu pomocí JavaScriptu popsaná výše není dostatečná a lze ji jednoduše obejít. Proto je PHP skriptem pomocí funkce shell_exec spuštěn Linuxový program file, pomocí něhož lze zobrazit informace o souborech a zjistit, zda nahraný soubor je opravdu video nebo ne. V dalším kroku se přistoupí ke kontrole kodeku.

Ke kontrole kodeku slouží program mediainfo spouštěný opět funkcí shell_exec. Tento program umožňuje výpis veškerých potřebných informací o multimediálním souboru. Pokud soubor není uložen ve správném kodeku nebo se nejedná o video, je soubor smazán.

Ke smazání souboru slouží funkce unlink. Aby bylo možné soubor smazat, musí mít správně nastavená přístupová práva. Přístupová práva se mění pomocí funkce chmod. Práva jsou u souboru změněna podle následujícího klíče: vlastník souboru společně se skupinou které soubor patří mají nastavena práva pro čtení a zápis, ostatní uživatelé pro čtení. v opačném případě je nutné vyčíst délku videa.

K vyčtení délky videa vhodně poslouží program ffmpeg opět spouštěný funkcí shell_exec. V tomto okamžiku je možné přikročit k zanesení záznamu o souboru do tabulky databázového systému.

S databázovým systémem je nutné nejprve navázat spojení. K tomu slouží funkce mysqli_connect. V případě úspěchu se pomocí MySQL direktiv a vyčtených informací uložených v jednotlivých proměnných vkládají záznamy do tabulky za pomoci funkce mysqli_query. V případě selhání funkce mysqli_connect nebo funkce mysqli_query je nutné vytvořit záznam o vzniklé chybě.

O vzniklých chybách informuje zvláštní textový soubor. Jednotlivé řádky v tomto souboru charakterizují vzniklé chyby, jednotlivé položky v řádce oddělené tabulátorem

informace, jež je nutné zanechat do databáze ručně. K textovému souboru se přistupuje pomocí funkcí `fopen` (pro otevření souboru a specifikaci práce se souborem) a `fwrite` (pro zápis do souboru). Nakonec je třeba soubor zavřít funkcí `fclose`. Aby bylo možné k souboru přistupovat pomocí PHP skriptu spuštěného uživatelem přes webové rozhraní, je nutné, aby měl soubor správně nastaveného vlastníka a skupinu, jinak PHP soubor neotevře z důvodů přístupových práv. Protože proces apache reprezentující server je vlastněn uživatelem `www-data`, jako vlastníka a skupinu souboru je nutné nastavit právě jeho [23]. K tomu je použit program `chown` s následující syntaxí:

```
sudo chown www-data:www-data logfileDB.txt
```

5.2.3 Sestavování vysílaného programu

Vysílaný program není nic jiného než posloupnost multimediálních souborů řazená za sebou uložená ve vhodném formátu. Program VLC, se kterým architektura počítá jako se streamovacím serverem, dokáže kontinuálně streamovat multimediální soubory uložené ve formátu `m3u`. Tento formát VLC využívá jako formát playlistu. Jedná se o textový soubor s následující strukturou:

```
#EXTM3U
#EXTINF:55,Medvedi.mp4
/home/dada/Videa/Medvedi.mp4
#EXTINF:113,Medvedi2.mp4
/home/dada/Videa/Medvedi2.mp4
```

První řádek označuje, že se jedná o soubor `m3u`. Další řádky specifikují jednotlivé multimediální soubory; číselný údaj za dvojtečkou udává délku souboru v sekundách, údaj za čárkou jméno souboru a následně na novém řádku je uvedena cesta k souboru. Zbytek podkapitoly se bude věnovat algoritmu a realizaci sestavování vysílacího programu.

Sestavování vysílacího programu se děje pravidelně a proto je nutné příslušný skript spouštět periodicky. K tomuto účelu slouží na UNIXových platformách nástroj `cron`.

Nástroj `cron` tedy jednou denně spustí obslužný `php` skript, který musí postupně provést následující úkony:

1. Připojit se k MySQL databázi
2. Vyfiltrovat z databáze schválená videa
3. Z vyfiltrovaných záznamů naplnit proměnou typu `array`
4. Na základě informace o zařazení jednotlivých videích a jejich délce sestavit vysílací program
5. Uložit vysílací program ve formě textového souboru s příponou `m3u`

6. Uzavřít připojení k databázi

Postup připojení k databázi byl již popsán u předchozích skriptů a proto bude v této podkapitole vynechán stejně jako popis uzavření připojení k databázi. Místo toho bude věnován prostor popisu zbylých úkonů.

1. Vyfiltrování schválených videí

Pomocí SQL direktivy SELECT lze z konkrétní tabulky databáze vyfiltrovat záznamy odpovídající určitým pravidlům. Konkrétní zápis vypadá následovně:

```
SELECT ID, Delka, Zarazeni FROM `upload_info` WHERE Schvaleno=1
```

Pomocí tohoto zápisu jsou z tabulky upload_info vybrány všechny záznamy (respektive informace o jejich ID, délce a zařazení) splňující podmínku Schvaleno=1. Tyto záznamy jsou uloženy v php proměnné pole typu array, která má následující strukturu:

```
pole: Array
(
  [0] => Array
    (
      [0] => 14_05_15_15_2018-5af98bdc2c20e
      [1] => 00:00:58
      [2] => C
    )

  [1] => Array
    (
      [0] => 19_04_13_11_2018-5ad8795b6a6b3
      [1] => 00:01:54
      [2] => A
    )

  [2] => Array
    (
      [0] => 27_05_18_02_2018-5b0ad6a828f9c
      [1] => 00:00:55
      [2] => B
    )
)
```

V tomto kroku je také možné napočítat celkový čas ve formátu hh:mm:ss, který schválená videa zabírají. Ten je uložen v proměnné cas typu array s následující strukturou:

```
Array
(
  [h] => 0
  [m] => 3
  [s] => 47
)
```

)
Z příkladu je patrné, že schválená videa jsou celkem tři a o každém je v proměnné pole uložena informace o ID, délce a zařazení. Toto pole je ovšem z hlediska zpracování nevhodně uspořádané a je výhodnější rozřídění podle zařazení o které se stará následující krok.

2. Přetřídění podle zařazení

Požadovaný formát ve kterém mají být záznamy o souborech uloženy z důvodu snazší manipulace a konstrukce kódu sestavujícího vysílaný program je následující:
zaznamy: Array

```
(  
  [A] => Array  
  (  
    [0] => Array  
    (  
      [stopaz] => 00:01:54  
      [name] => 19_04_13_11_2018-5ad8795b6a6b3  
    )  
  )  
)  
  
[B] => Array  
(  
  [0] => Array  
  (  
    [stopaz] => 00:00:55  
    [name] => 27_05_18_02_2018-5b0ad6a828f9c  
  )  
)  
  
[C] => Array  
(  
  [0] => Array  
  (  
    [stopaz] => 00:00:58  
    [name] => 14_05_15_15_2018-5af98bdc2c20e  
  )  
)  
  
[D] => Array  
(  
)  
  
[0] => Array  
(  
)  
)
```

Z příkladu je na první pohled patrné, kolik existuje schválených videí v jednotlivých kategoriích, jejich stopáž a nakonec i jméno souboru. Tyto informace společně tvoří vše potřebné pro sestavení vysílaného programu.

3. Sestavení vysílacího programu

Následujícím krokem je řazení videí za sebe podle programového schématu a vytvoření tak vysílaného programu. Posloupnost jmen vysílaných videí je uložena v proměnné PROGRAM typu pole. Algoritmus této části skriptu pro naplnění vysílacího okna pro zařazení A je následující:

1. nastav pomocnou proměnnou čas = 0
2. opakuj následující kroky dokud čas != délka časového okna +/- tolerance
3. vygeneruj náhodné číslo X v rozsahu 0 až počet prvků pole zaznamy=>A
4. k proměnné čas přičti stopáž záznamu zaznamy=>A=>X=>stopaz
5. na konec proměnné program přidej zaznamy=>A=>X=>name (pomocí funkce array_push)
6. smaž z pole zaznamy záznam zaznamy=>A=>X a přeindexuj (pomocí funkce array_splice)

5.3. Uživatelské rozhraní

Uživatelské rozhraní má umožnit především jednoduchou orientaci v systému internetové televize a zajistit uživatelům potřebné funkcionality. Protože je televizní vysílání už od začátku specifikováno jako internetové v síti FEL ZČU, je logickým řešením vytvořit uživatelské rozhraní jako webové pomocí HTML.

Protože existují dvě skupiny s různým stupněm oprávnění, je nutné pro každou z nich vytvořit vlastní uživatelské rozhraní, umožňující jim provádět příslušné operace. Systém specifikuje dvě skupiny uživatelů podle úrovně oprávnění, jsou to: reportér a redaktor.

5.3.1 Reportér

S právy reportéra přistupuje k internetové televizi každý student FEL ZČU přihlášený přes Orion konto. K oprávněním této skupiny náleží nahrávat vlastní autorský obsah na stranu serveru.

Uživatelské rozhraní musí tedy především umožnit tuto funkcionalitu. Klient se připojuje metodou unicast za pomoci protokolu HTTP k serveru a v případě, že se rozhodne k uploadu obsahu, posílá na určitý port jednotlivé pakety vybraného souboru. Server je přijímá a dále zpracovává. Uživatelské rozhraní je tedy webová stránka napsaná

jazykem HTML umožňující nahrání souboru a zobrazující nejnútější informace o projektu internetové televize.

5.3.2 Redaktor

Redaktor má možnost schvalovat videa k vysílání. Jeho uživatelské rozhraní tedy musí přehledným způsobem poskytovat informaci o nahraném multimediálním obsahu na serverové straně. Jako nejsrozumitelnější forma se jeví tabulka a proto návrh počítá s touto formou zobrazení.

Řádky tabulky představují jednotlivá autorská videa, sloupce potom příslušné informace. Obsah tabulky je shodný s obsahem tabulky nahraného obsahu v databázovém systému. V případě, že reportér uzná určité video za vhodné k vysílání, upraví v tabulce příslušný atribut a tabulku uloží tlačítkem „Aktualizuj“. V případě, že redaktor změní určité atributy a posléze se rozhodne, že je chce vrátit na původní hodnotu, má možnost zmáčknutím tlačítka „Vrat' změny“ tak učinit. To je možné, pokud tabulka již nebyla aktualizována. Redaktor také může videa mazat, popřípadě k nim přidávat komentáře. Ve všech jmenovaných případech se jedná o komunikaci s databázovým systémem, která je v režii skriptovacího jazyka PHP. Aby redaktor mohl rozhodnout o tom, zda obsah bude nebo nebude schválen, je nutné, aby ho předem zhlédl. K tomuto účelu je v tabulce navíc sloupec označen jako „kouknout“, jehož obsahem je tlačítko, po jehož zmáčknutí se dané video přehraje v okně prohlížeče. Zobrazené záznamy tabulky je dále možné řadit vzestupně nebo sestupně podle klíčů: „ID“, „Původní název“, „Kdy byl nahrán“, „Schválen“, „Neschválen“ a „Zařazení“. Rozhraní dále poskytuje informaci o celkovém množství nahraného obsahu, celkovém množství schválených, neschválených a nezkontrolovaných videí. V dalším textu bude hrubě popsána struktura kódu, detailní popis je uveden na konci podkapitoly.

Kód se skládá z části napsané v javascriptu, HTML a PHP. HTML část kódu má na starost zobrazování tabulky, odkazů a definuje styly. Javascriptové funkce obsluhují události vyvolané zmáčknutím některého tlačítka. PHP část má na starost komunikaci s databázovým systémem pomocí SQL direktiv, přidělování na základě vyčtených informací jednotlivým HTML tagům unikátní id, popřípadě unikátní hodnotu atributu value. PHP podle hodnoty sloupce schváleno dále rozhoduje o zbarvení jednotlivých řádků tabulky. V případě zobrazování souhrnného přehledu o počtu nahraných, schválených, neschválených a nezkontrolovaných videí PHP zobrazuje podle toho, zda je uvedený počet nulový nebo

ne, buď prostý text nebo odkaz. V případě, že je zobrazen odkaz, je možné na uvedený údaj kliknout a tím v tabulce odfiltrovat žádaná videa nebo záznamy v tabulce seřadit.

5.3.3 Divák

V případě, že chce divák internetovou televizi pouze sledovat, postačí mu k tomu program VLC, v němž otevře příslušný síťový proud.

5.4. VLC

O spouštění programu se stará plánovač úloh (cron), který jednou denně v jasně stanovenou dobu spustí následující příkaz:

```
cvlc program.m3u --sout='#rtp{dst=127.0.0.1,port=5004,mux=ts,caching=2000,raw}' vlc://quit
```

1. Spustí se instance programu VLC bez grafického rozhraní
2. soubor program.m3u (playlist) se začne streamovat rtp protokolem na adresu dst, na port číslo 5004, je použit protokol MPEG-TS
3. vlc://quit ukončí instanci programu VLC po skončení posledního vysílaného videa

6. Programové schéma

Programové schéma člení vysílací čas do jasně stanovených bloků, definuje formát pořadů a udává, kdy mají být které multimediální soubory vysílány. Návrh se zaměřuje na vytvoření jednotné a neměnné struktury nezávislé na dni, ročním období atp.

V kapitole je vysílací doba systematicky rozdělena od nejhrubšího po nejjemnější časový formát. V hrubém rozdělení je stanoven hlavní vysílací čas a čas určený pro reprízy; tyto dva formáty se dále dělí na kratší časové úseky, které tvoří základní kameny pro tvorbu programu a které se dále zjemňují až na základní vysílací formáty.

V závěru kapitoly je uvedeno celkové shrnutí všech poznatků vyplývajících z jednotlivých podkapitol.

6.1. Hrubé rozdělení

Cílem návrhu je vytvořit programové schéma pro jeden den, nejdelší formát je tedy 24 hodinový. Tento formát je rozdělen na dvě stejně dlouhá časová okna o délce 12 hodin; jedno je určeno pro hlavní vysílání a druhé pro reprízy. Architektura počítá s tím, že pořad vysílaný v hlavním vysílacím čase může být ve stejný den přehrán ještě jednou v čase určeném pro reprízy.

V následujících podkapitolách bude věnována pozornost pouze hlavnímu vysílacímu času, bude provedeno jeho jemnější dělení až na základní vysílací formáty.

6.2. Hlavní vysílací čas

Délka hlavního vysílacího času je tedy 12 hodin. Tento formát je rozdělen na čtyři nestejně dlouhé tématické (žánrové) okruhy označené písmeny A,B,C,D. Z těchto tématických okruhů volí uživatel nahrávající vlastní multimediální obsah na serverovou stranu. Zadává tak do databázového systému informaci o tom, kdy si myslí, že by bylo nejvhodnější jeho video vysílat.

Požadavek na strukturu 12ti hodinového formátu je jednoduchost. Z tohoto důvodu jsou zvoleny dva subformáty z nichž je složen hlavní vysílací čas. První subformát má délku 90 minut a druhý 180 minut. 90ti minutový subformát je použit dvakrát, 180ti minutový třikrát.

Z důvodu obtížné manipulace s 12ti hodinovým časovým formátem je tento rozdělen na dva 6ti hodinové. Jemnější dělení 6ti hodinového formátu pak dává vzniknout základním vysílacím formátům.

Cílem dělení na kratší časové formáty je jednodušší manipulace a plnění vysílaným obsahem, vytvoření opakovatelného schématu užitého při vytváření vysílaného programu a jednodušší programová konstrukce kódu vytvářejícího vysílaný program (popsáno v kapitole 5.2.3)

Výsledkem dělení hlavního vysílacího času je tedy 12ti hodinový formát složený z formátu 90ti a 180ti minutového, který je dále rozdělen na 6ti hodinový formát.

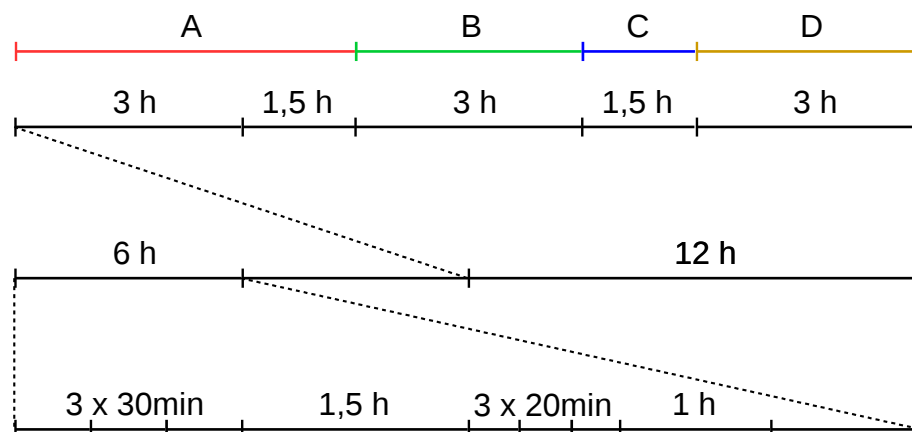
6.2.1 Dělení na základní vysílací formáty

Jemnějším dělením 6ti hodinového formátu vznikají formáty základní; ty jsou celkem čtyři: 20ti minutový, 30ti minutový, 60ti minutový a 90ti minutový.

V 6ti hodinovém okně je 20ti minutový formát použit 3krát, 30ti minutový rovněž, 60ti minutový je použit dvakrát a 90ti minutový jednou.

6.2.2 Shrnutí

Následující obrázek graficky znázorňuje výše popsané dělení vysílacího času. Barevně jsou označeny žánrové okruhy A, B, C, D, pod nimi je uvedeno jakou zabírají vysílací dobu a jak jsou řazeny za sebou v hlavním vysílacím čase. Ve spodní části obrázku je uvedeno rozdělení 6ti hodinového formátu na formáty základní.



Obrázek 2: Vysílací formáty

6.3. Tvorba denního programu

Tato podkapitola popíše jakým způsobem je tvořen denní vysílací program. Vysílací program je budován na základě dříve popsaných časových formátů ke kterým navíc přibude několik formátů dalších sloužících pro předěly mezi jednotlivými tématickými okruhy.

Výsledkem je finální podoba denního programového schématu s jasně stanovenými dobami začátků a konců vysílání jednotlivých tématických (žánrových) okruhů.

Na konci podkapitoly je opět uvedeno shrnutí společně s grafickou reprezentací.

6.3.1 Předěly a jejich formáty

Předěly od sebe oddělují jednotlivé vysílací celky (pořady, okruhy atp.) Jejich úlohou je dát dennímu vysílacímu programu rozumný řád ve smyslu začátků a konců jednotlivých vysílaných úseků.

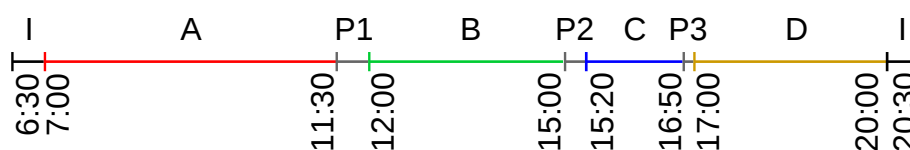
Z důvodů které vyplynou dále jsou stanoveny následující tři časové formáty předělů: 1x 10ti minutový, 1x 20ti minutový a 1x 30ti minutový. Návrh počítá s použitím každého předělu jednou; celková vysílací doba se tedy s použitím předělů prodlouží o hodinu.

6.3.2 Denní informace

Čas pro denní informace je rezervován pro potřeby sdělit ve vysílacím čase zásadní denním aktuality. Návrh k tomuto účelu stanovuje 30ti minutový formát, který je vložen před hlavní vysílací čas a jednou se opakuje na konci vysílání.

Celková vysílací doba se tak s denní informací prodlouží o další hodinu.

6.3.3 Shrnutí



Obrázek 3: Programové schéma

7. Závěr

Práce v úvodu objasňuje motivace a cíle projektu a následně vysvětluje důležité pojmy z oblasti streamování multimédií internetovou sítí. Vysvětleny jsou pojmy jako server, kodek a podobně.

V další kapitole je nastíněna obecná představa o fungování systému. Jsou stanoveny jeho možnosti a funkce.

Následující kapitola popisuje možnost realizace cílů stanovených v předchozí kapitole. Jsou vybrány konkrétní nástroje pro řešení problematiky.

Nástroje jsou spojeny v funkční celky, jejichž fungování a realizace je popsána v 5. kapitole.

6. kapitola se věnuje problematice programového schématu. Jsou vytvořeny základní vysílací formáty a denní programové schéma.

Test systému byl proveden lokálně. Ověřené funkcionality jsou:

1. upload souboru + kontrola
2. zanášení záznamů do databázového systému
3. dynamické vytváření vysílaného programu
4. multimediální stream na adresu localhostu
5. uživatelské rozhraní pro obě skupiny uživatelů

Seznam použitých zkratek

AJAX	Asynchronous Javascript and XML
AVI	Audio Video Interleave
DVB-C	Digital Video Broadcasting – Cable
DVB-S	Digital Video Broadcasting – Satellite
DVB-T	Digital Video Broadcasting – Terrestrial
FTP	File Transfer Protocol
GNU/GPLv2	GNU General Public License
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IPTV	Internet Protocol Television
LAMP	Linux, Apache, MySQL, PHP
MIME	Multipurpose Internet Mail Extensions
MPEG	Motion Pictures Experts Group
PHP	Hypertext Preprocessor
PS	Program Stream
RTP	Real Time Transport Protocol
SQL	Structured Query Language
TS	Transport Stream
VOB	Video Object

Seznam literatury a informačních zdrojů

- [1] **Statut české televize** > článek II Veřejná služba české televize [online] 2018
Dostupné z: <http://www.ceskatelevize.cz/vse-o-ct/statut-ct/>
- [2] **Rada pro rozhlasové a televizní vysílání**. In: *RRTV* [online]. [cit. 2018-06-11].
Dostupné z: <https://www.rrtv.cz/cz/static/o-rade/poslani-rady/index.htm>
- [3] **Televizní vysílání**. In: *Wikipedia* [online]. Wikimedia Foundation [cit. 2018-06-11].
Dostupné z: https://cs.wikipedia.org/wiki/Televizn%C3%AD_vys%C3%ADl%C3%A1n%C3%AD
- [4] **Zákon č. 111/1998 Sb část první §1**
- [5] **Televize**. In: *Wikisofia* [online]. 2013 [cit. 2018-06-11]. Dostupné z:
https://wikisofia.cz/wiki/Televize#cite_note-Osvaldov.C3.A1-2
- [6] **Tenky vs tlusty klient**. In: *Abclinuxu* [online]. 2009 [cit. 2018-06-11]. Dostupné z:
<http://www.abclinuxu.cz/poradna/linux/show/279978>
- [7] **Server**. In: *Adaptic* [online]. [cit. 2018-06-11]. Dostupné z:
<http://www.adaptic.cz/znalosti/slovnicek/server/>
- [8] **Streaming media**. In: *Techtarget* [online]. 2009 [cit. 2018-06-11]. Dostupné z:
<http://whatis.techtarget.com/definition/streaming-media>
- [9] **Data stream**. In: *Techopedia* [online]. [cit. 2018-06-11]. Dostupné z:
<https://www.techopedia.com/definition/6757/data-stream>
- [10] **Protokol**. In: *It-slovník* [online]. [cit. 2018-06-11]. Dostupné z: <https://it-slovník.cz/pojem/protokol>
- [11] **Protokol (informatika)**. In: *Dbpedia* [online]. [cit. 2018-06-11]. Dostupné z:
[http://cs.dbpedia.org/page/Protokol_\(informatika\)](http://cs.dbpedia.org/page/Protokol_(informatika))
- [12] **Historical trends in the usage of web servers**. In: *W3techs* [online]. [cit. 2018-06-11].
Dostupné z: https://w3techs.com/technologies/history_overview/web_server
- [13] **Základy relačních databází**. In: *Vse* [online]. [cit. 2018-06-11]. Dostupné z:
<https://gml.vse.cz/data/oppa-webdesign/zaklady-db.html>
- [14] **MySQL**. In: *Oracle* [online]. [cit. 2018-06-11]. Dostupné z:
<https://www.oracle.com/cz/mysql/index.html>
- [15] **PHP základy**. In: *Tvorba-webu* [online]. [cit. 2018-06-11]. Dostupné z:
<https://www.tvorba-webu.cz/php/>
- [16] **What can PHP do?** In: *Php.net* [online]. [cit. 2018-06-11]. Dostupné z:
<http://php.net/manual/en/intro-whatcando.php>
- [17] **PlexTv**. In: *Plex.tv* [online]. [cit. 2018-06-11]. Dostupné z: <https://www.plex.tv>
- [18] **FFmpeg**. In: *Ffmpeg.org* [online]. [cit. 2018-06-11]. Dostupné z: <http://ffmpeg.org>
- [19] **Wowza**. In: *Wowza.com* [online]. [cit. 2018-06-11]. Dostupné z:
<https://www.wowza.com/>
- [19] **Description of core php.ini directives**. In: *Php.net* [online]. [cit. 2018-06-11].
Dostupné z: <http://php.net/manual/en/ini.core.php>
- [20] **Velikost nahrávaných souborů**. In: *Vrana.cz* [online]. 2005 [cit. 2018-06-11].
Dostupné z: <https://php.vrana.cz/velikost-nahravanych-souboru.php>
- [21] **HTML <input> Tag**. In: *W3school* [online]. [cit. 2018-06-11]. Dostupné z:
https://www.w3schools.com/tags/tag_input.asp
- [22] **File Upload Progress Bar Meter Tutorial Ajax PHP**. In: *Developphp* [online].
[cit. 2018-06-11]. Dostupné z: <http://www.developphp.com/video/JavaScript/File-Upload-Progress-Bar-Meter-Tutorial-Ajax-PHP>

- [23] **File Permission Error in PHP fopen() function.** In: *Linuxquestions* [online]. 2003 [cit. 2018-06-11]. Dostupné z: <https://www.linuxquestions.org/questions/linux-software-2/file-permission-error-in-php-fopen-function-80533/>