VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ
DEPARTMENT OF INTELLIGENT SYSTEMS

MOBILNÍ KAMERA REALIZOVANÁ PROSTŘEDKY ROS2

THESIS TITLE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **DANIEL ONDERKA**
AUTHOR

VEDOUCÍ PRÁCE **doc. Ing. VLADIMÍR JANOUŠEK, Ph.D.**
SUPERVISOR

BRNO 2023

Abstrakt

Do tohoto odstavce bude zapsán výtah (abstrakt) práce v českém (slovenském) jazyce.

Abstract

Do tohoto odstavce bude zapsán výtah (abstrakt) práce v anglickém jazyce.

Klíčová slova

Sem budou zapsána jednotlivá klíčová slova v českém (slovenském) jazyce, oddělená čárkami.

Keywords

Sem budou zapsána jednotlivá klíčová slova v anglickém jazyce, oddělená čárkami.

Citace

ONDERKA, Daniel. *Mobilní kamera realizovaná prostředky ROS2*. Brno, 2023. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce doc. Ing. Vladimír Janoušek, Ph.D.

Mobilní kamera realizovaná prostředky ROS2

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana X... Další informace mi poskytli... Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Daniel Onderka
18. prosince 2023

Poděkování

V této sekci je možno uvést poděkování vedoucímu práce a těm, kteří poskytli odbornou pomoc (externí zadavatel, konzultant apod.).

Obsah

Seznam obrázků

Kapitola 1

Úvod

Kapitola 2

Teoretická část

2.1 Hardware Adeept AWR 4WD a Raspberry Pi

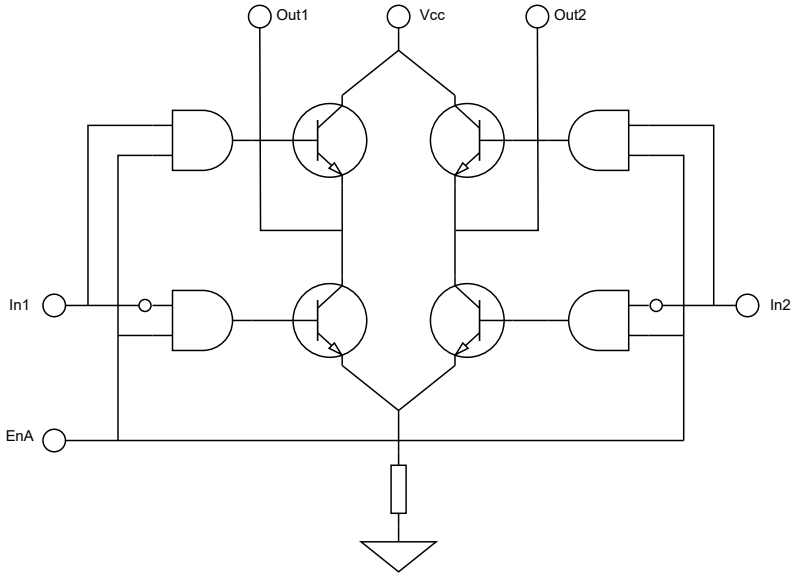
Celá práce je implementována nad existujícím hardwarem. Konkrétně se jedná o "stavebnici"Adeept AWR 4WD z níž pochází všechny periferie, jako motory, serva a čidla. Tyto periferie jsou pak ovládány mikropočítačem Raspberry Pi4. V porovnání s běžně používanými mikrokontrolery se jedná o výkonější hardware.

Adeept AWR 4WD

Robot HAT

Adeept Robot HAT je hardwarová deska, která slouží k rozšíření Rapsberry Pi o další funkcionalitu. K rpi se připojuje pomocí GPIO(General purpuse input outpu) pinů. Deska jako taková obsahuje rozšiřující čipy a rozhraní sloužící k ovládání připojených periferií.

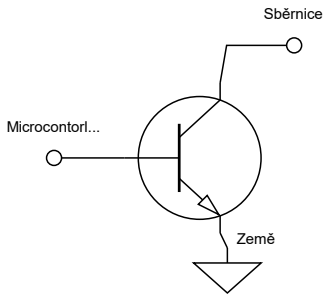
- PCA9685 [2]
  - generátor PWM signálu
  - 16 kanálů
  - 12 bitů rozlišení střídy (4096 možných hodnot)
  - je ovládaný přes I2C sběrnici
- L298P
  - ovladač pro řízení dc motoru
  - obsahuje full bridge obvod (viz následující obrázek)
  - umožňuje roztočit motor oběma směry
  - pomocí PWM lze ovládat rychlost motorů
- další rozhraní pro připojení periferií (ultrasonic sensor, line tracking)



Obrázek 2.1: Full bridge konfigurace pro ovládání motoru. In1 a In2 určují směr otáčení. EnA je PWM signál určující rychlost otáčení. [7]

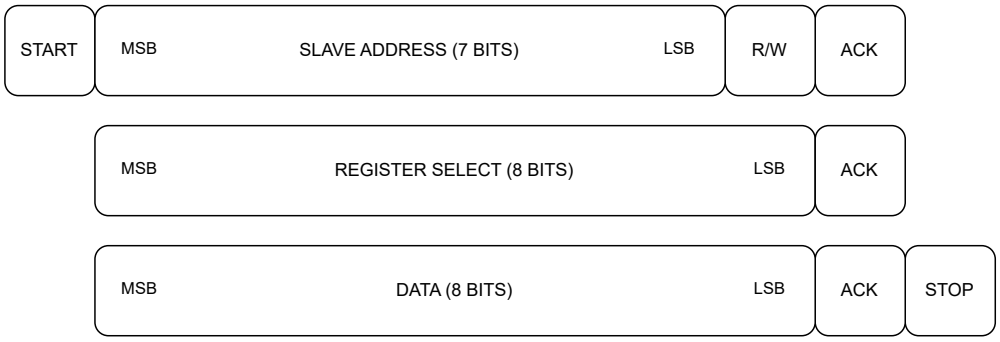
I2C

Je synchronní sběrnice, která se vyznačuje svou jednoduchostí a nízkou cenou. Využívá dva vodiče SDA (serial data) a SCL (serial clock). Oba vodiče jsou připojeny k napájecímu napětí pomocí pull-up rezistoru a bez vlivu jiného hardwaru zůstávají v logické jedničce. Zařízení která jsou na tuto sběrnici připojeny využívají open drain k úpravě aktuální napěťové úrovně na sběrnici. I2C pak pracuje s dvěma druhy zařízení, master a slave. Master zahajuje a ukončuje komunikaci, typicky se jedná o mikrokontroler. Slave jsou pak ostatní zařízení s nimiž může master komunikovat, typicky různé periferie. [1]



Obrázek 2.2: Open drain

Přenos jednoho datového rámce zahájí master zařízení přivedením datové sběrnice do nuly. Následující komunikace se skládá z odeslání rámce o délce osmi bitů a potvrzení o úspěšném přenosu dat od přijímajícího zařízení. Toto potvrzení se nazývá ACK a je provedeno podržením datové sběrnice v hodnotě nula po dobu jednoho taktu. Opačný stav se nazývá NACK a indikuje že došlo k chybě. Ukončení přenosu je provedeno navrácením datové sběrnice na hodnotu jedna.

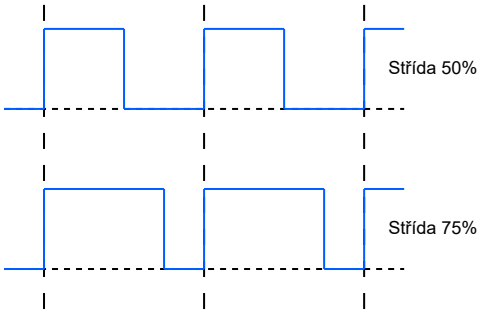


Obrázek 2.3: Na obrázku lze vidět, jak může vypadat přenos jednoho datového slova. V prvním rámci je přenesena sedmi bitová adresa sloužící k výběru slave zařízení se kterým chce máster navázat komunikaci. Doplněná o jeden bit indikující směr, kterým potečou data. V druhém rámci dojde k adresaci konkrétního registru a ve třetím se pošlou samotná data. [6]

Pulzně šířková modulace

Jedná se o techniku která umožňuje vytvořit pseudo-analogový výstup s použitím digitálních pinů. Mikrokontrolery jsou digitální zařízení a chtěly by tedy s okolním světem komunikovat pomocí digitálních pinů. Reálný svět však nefunguje pouze na úrovni jedniček a nul a je poroto potřeba převádět výstup z mikrokontroleru na analogový signlál. Problém je v tom, že převod digitálního signálu na analogový je relativně dlouhá a neefektivní operace. Proto vznikla pulzně šířková modulace, která umožňuje relativně jednoduše simulovat analogový výstup.

Při pohledu na klasický digitální signál který rovnoměrně střídá vysokou a nízkou úroveň by šlo říci, že se jedná o PWM signál se střídou 50%. Střída udává poměr času kdy je signál v logické jedničce ku času kdy je v nule. Součet těchto hodnot se musí rovnat délce jedné periody. Úpravou tohoto poměru lze simulovat analogový signál.



Obrázek 2.4: Signál pro různé hodnoty střídy

DC Motor

Pohyb celého autíčka zajišťují čtyři motory. Jsou napájeny pomocí stejnosměrného proudu(dc - direct current). K jejich ovládání a dodání většího proudu než je schopno poskytnout raspberry pi slouží obvod L298P který je součástí Robot HAT. Elektrický DC motor se skládá ze dvou hlavních částí, stator a rotor. Stator je statická, vnější část, a typicky se

jedná o permanentní magnet. Uvnitř statoru se pak nachází rotor, ten se skládá z elektromagnetů, které při zapnutí reagují s statorem(póly se odpuzují a přitahují) a dojde tak k částečnému pootočení. Při správném spínání a vypínání těchto magnetů lze motor rozběhnout. To zajišťuje komutátor, jedná se o prstenec složený z několika navzájem odizolovaných částí, které jsou připojeny na vývody elektromagnetu. S povrchem prstence jsou pomocí pružin v kontaktu dva kartáče, které jsou připojeny na napájení a zemi. Komutátor se otáčí společně s rotorem a při tomto pohybu se kartáče postupně dotýkají různých částí komutátoru a spínají tak jednotlivé elektromagnety, ty vedou k pootočení rotoru a sepnutí následujícího magnetu.

Obrázek 2.5: Signál pro různé hodnoty střídy

Dc motor má stator, permanentní magnet. A rotor, cívky do kterých se pak posílá proud tak aby se od statoru odpuzovaly a roztáčí se tak rotor.

Led

Servo

Adeept AWD 4WD využívá pouze jedno servo, a to na ovládání úhlu kamery. Konkrétně se jedná o model Adeept AD002. Je ovládané pomocí PWM signálu. Generování tohoto signálu zajišťuje čip PCA9685 který je umístěný na Robot HAT.

Depth module

Funguje na principu radaru. Vyšle zvukovou vlnu na frekvenci 40Khz a uloží si časovou značku. Následně poslouchá než se mu vlna vrátí a tehdy uloží druhou značku. Kdyz pak ví jak rychle se šíří zvuk ve vzduchu dokáže vypočítat vzdálenost od překážky.

$$S = (T_2 - T_1) * V_S/2$$

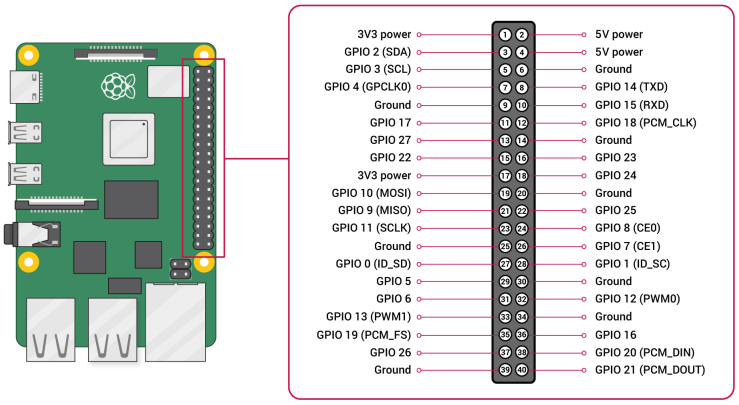
Kde T_1 je moment kdy byla vyslána vlna T_2 kdy byla vlna přijata a V_S rychlost šíření zvuku ve vzduchu.

Line tracking

Funguje podobne jako depth module. Vysílá ultračervené tentokrát světlo a sleduje jestli došlo k odrazu. Bílý papír většinu světla odrazí a senzor jej zachytí, naopak černá čára světlo pohltí a senzor tak pozná že našel čáru.

Raspberry Pi 4b

Jako mozek celého systému je použit mikropočítač Raspberry Pi. Konkrétně se jedná o verzi 4 model B s operační pamětí o velikostí čtyř gigabajtů. Tato verze obsahuje 64bitový procesor, který je potřeba pro spuštění 64 bitového Ubuntu serveru, který je doporučeným operačním systémem pro požití ROS2 na raspberry pi. Komunikace s většinou použitých periferií je uskutečněna pomocí General Purpuse Input Output(GPIO) pinů. Jedná se o digitální vývody, které podle potřeby můžou fungovat jako vstup i výstup ze zařízení. Některé z nich pak mají ještě speciální funkce, například GPIO 2 a 3 můžou pracovat jako SDA a SCL připojení pro I2C komunikaci.



Obrázek 2.6: GPIO pinout

Camera

2.2 Aktuální software

Tento robot je dodáván s ukázkovým softwarem. Ten je implementován v jazyku python a využívá knihovny třetích stran sloužící k nízkoúrovňovému ovládání hardware komponent. Aby byl robot responzivní je celá implementace řešena s použitím python modulů sloužících k realizaci multithreadingu. Tento přístup bezpochyby funguje, avšak i u tohoto relativně malého projektu se takový kód stává poměrně složitým na porozumění.

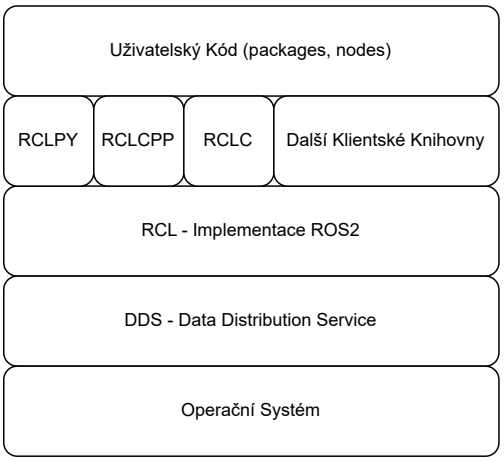
2.3 Seznámení s ROS2

ROS2 je middleware sloužící k vývoji a řízení robotů. Middleware je softwarová vrstva běžící nad operačním systémem sloužící k rozšíření jeho funkcionality. Typicky middleware obsahuje knihovny, ovladače, vývojové a monitorovací nástroje. Může také specifikovat doporučené metodologie pro vývoj. ROS2 je již druhá verze tohoto softwaru, která rozšiřuje a opravuje neduhy první verze. Původní ROS1 je považován za de-facto standart pro vývoj robotických aplikací. Tato práce využívá ros2 distribuci jménem iron. Distribuce v ROS2 lze popsat jako set operačního systému, knihoven a dalších aplikací, které jsou otestovány a je zaručeno, že jsou navzájem kompatibilní. Velkou výhodou ROS je fakt, že se jedná o open source projekt. Díky tomu kolem něj vznikla velká komunita vývojářů, ale i firem a dalších institucí, které tvoří mnoho souvisejícího obsahu. Existuje tedy velké množství knihoven, dokumentací a návodů které usnadňují vývojářům práci.

Vrstvy ROS2

Na nejvyšší úrovni, se nachází programátor který interaguje s klientskými knihovnami pro vývoj ROS2 aplikací. Tyto knihovny jsou oficiálně dvě a to rclpy pro python a rclcpp pro C++. Existují také implementace pro další programovací jazyky (rclc, java, C#), ty jsou však udržovány komunitně. Všechny klientské knihovny pak využívají rcl. RCL je jádro celého ROS implementované v jazyce C. RCL implementuje hlavní ros2 funkcionalitu, kte-

rou následně poskytuje pomocí rozhraní jednotlivým klientským knihovnám. Díky tomuto se funkcionalita implementovaná v pythonu bude chovat stejně jako ta implementovaná v c++. Taky pak kód implementovaný v c++ může komunikovat s tím v pythonu. Poslední vrstvou je data distribution service. DDS je komunikační vrstva implementována na UDP protokolu sloužící k předávání informací mezi procesy. Má charakteristiky systémů reálného času, zajišťuje kvalitu a zabezpečení komunikace. Také umožňuje vyhledávání uzlů bez potřeby centralizovaného serveru (využívá k tomu multicast, komunikace mezi uzly je následně unicast).



Obrázek 2.7: Layers

2.3.1 Vývoj v ROS2

Nejvyšší organizační jednotkou v ROS2 je workspace. Jedná se o složku, která slouží k organizaci zdrojových souborů, jejich instalaci a následné spouštění. ROS2 instalace je také workspace a před použitím je potřeba ji nejprve aktivovat. K tomu v linuxu slouží příkaz `source`. Aktivace workspace je akumulativní a v jeden moment tedy může být aktivních několik workspace. Typicky se první aktivuje základní ros2 instalace, která tvoří takzvanou underlay vrstvu. Vývojový workspace aktivovaný jako druhý se pak nazývá overlay. Pokud má overlay nějaké závislosti, měly by být uspokojeny v underlay. Zdrojové soubory v rámci workspace jsou pak organizovány do packages. Package může obsahovat zdrojové soubory,

knihovny a definice zpráv. Packages na sobě můžou navzájem záviset (např: package která potřebuje interface závisí na jiné která tento interface definuje).

```
Workspace
├─ build       #soubory používané při kompilaci
├─ install     #výsledky kompilace a další soubory potřebné ke spuštění
├─ log         #logy z kompilace
├─ launch      #launch soubory
└─ src         #packages
   └─ package_name     #příklad jak vypadá python package
      ├─ package_name     #zdrojové soubory
      ├─ resource
      ├─ test
      ├─ package.xml   #metadata o package
      ├─ setup.cfg     #pro ros2 run (spouštění package)
      └─ setup.py      #instrukce pro kompilátor jak nainstalovat package
```

Node

Celý ROS2 systém je složený z uzlů (Node), které mezi sebou navzájem komunikují. Každý uzel je vlastní výpočetní jednotka, která by měla plnit jeden specifický úkol. Tento přístup je podobný objektově orientovanému návrhu a ROS2 jej také využívá. Implementačně je tedy uzel objekt který dědí ze třídy `Node`. Uzly v ROS2 většinou nepotřebují běžet permanentně, ale pouze v monentě, kdy nastane nějaká událost kterou je potřeba obsloužit. Z toho důvodu existuje v ROS2 metoda `spin()`, která uspí vykonávání uzlu, dokud jej není potřeba opět využít. Aby ROS2 šetřil výpočetní prostředky, využívá dva přístupy k určení, kdy bude potřeba uzel vzbudit. Prvním je iterative execution, ten se používá u uzlů, které vykonávají svou činnost pravidelně na nějaké frekvenci. Například se může jednat o výpočetní uzel, který pravidelně každých x mikrosekund provede výpočet podle hodnot senzorů a odešle výsledek. Druhý je event oriented execution. Zde dochází k vyvolání řídícího cyklu jako důsledek nějaké události, typicky se jedná od příchozí zprávu z subscription, service nebo action. Frekvenci spouštění těchto uzlů pak lze odvodit od frekvence příchozích zpráv. Typicky se může jednat o uzel přijímající snímky z kamery na kterých provede výpočet a vrátí odpověď. Frekvence výpočtu je dána příchozími snímky, pokud snímky přestanou přicházet, uzel se nebude spouštět.

Topic

Je základním a také nejčastěji používaným způsobem pomocí kterého spolu ROS2 uzly komunikují. Topic si lze představit jako analogii hardwarové sběrnice. Prakticky se jedná o přesně pojmenované místo, do kterého může n uzlů posílat data (Publish) a m poslouchat

Algoritmus 1: Struktura a použití Node objektu

```
1: class CustomNode(Node):
2:     def __init__(self):
3:         super().__init__('node_name')
4: def main(args):
5:     rclpy.init(args=args)
6:     node = CustomNode()
7:     rclpy.spin(node)
8:     node.destroy_node()
9:     rclpy.shutdown()
```

co bylo posláno (Subscribe). Zprávy posílané do topicu mají přesný formát určený pomocí Interface a jsou posílány asynchronně. Typickým příkladem použití může být topic, do nějž posílá data uzel ovládající kameru a několik dalších uzlů které tyto data potřebují jej mohou číst. [4]

Algoritmus 2: Subsrciber Node

```
1: self.create_subscription(Interface, "topic_name",
   self.callback_function, queue_size)
2: def callback_function(self, msg):
3:     value = msg.item
```

Tento kód ukazuje, jak se může uzel přihlásit v odebírání zpráv z topicku. Nejprve je potřeba (typicky v konstruktoru třídy) zavolat zděděnou metodu sloužící k inicializaci nějaké ROS2 funkcionality. V tomto případě se jedná o `create_subscription`. Jako parametry potřebuje jméno, interface, callback funkci a délku fronty. Inteface definuje formát zpráv a délka fronty je použita v případě, že uzel nezvládá přijímat zprávy dostatečně rychle. Callback funkce je pak zavolána pokaždé když do topicku přijde nová zpráva. Druhým parametrem callback funkce je předán objekt, který ve svých atributech obsahuje hodnoty dané zprávy.

Algoritmus 3: Publisher Node

```
1: self.publisher = self.create_publisher(Interface, "topic_name",
   queue_size)
2: output = Interface()
3: output.item = some_value
4: self.publisher.publish(output)
```

Odesílání zpráv do topicu demonstruje tento kód. Jeho struktura je podobná předchozímu příkladu. Odeslání zprávy demonstrují řádky 2–4. Nejprve dojde k inicializaci objektu interface, stejného datového typu jako ten, který používá topic. Tento objekt je následně naplněn daty a pomocí metody `publish()` předem vytvořeného publisheru odeslán.

Service

Sevice funguje stejně jako klient – server komunikace známá z počítačových sítí . Jedná se tedy o synchronní komunikaci kde jedna node poskytuje nějakou službu a ostatní si na ni mohou poslat požadavek. Od service se typicky předpokládá okamžitá odpověď aby nedošlo k narušení (control cycle) volajícího uzlu. [4]

Algoritmus 4: Service server

```
1: self.srv = self.create_service(Interface, "service_name",
   self.callback_function)
2: def callback_function(self, request, response):
3:     value = request.item
4:     response.item = some_value
5:     return response
```

Změnou oproti předchozím příkladům je přidání nového parametru do callback funkce. Tato funkce má dva důležité parametry, request a response. Request obsahuje konkrétní hodnoty požadavku na server a response je potřeba naplnit výsledky a vrátit z funkce.

Algoritmus 5: Service client

```
1: self.cli = self.create_client(Interface, "service_name")
2: while not self.cli.wait_for_service(timeout_sec=1.0):
3:     pass
4: def send_request(self):
5:     self.req = Interface.Request()
6:     self.req.item = some_value
7:     self.future = self.cli.call_async(self.req)
8:     rclpy.spin_until_future_complete(self, self.future)
9:     response = self.future.result()
10:    value = response.item
```

Service client narozdíl od topic subscriberu závisí na tom, aby existoval server, který je schopný odpovídat na jeho požadavky. Tato podmínka vyplývá z faktu, že service server by měl odpovídat na dotazy téměř okamžitě a service klient tedy předpokládá, že vždy dostane odpověď. Pokud by neexistoval server, klient by při požadavku skončil v nekonečném čekání. Proto je hned v konstruktoru implementována kontrola, které nedovolí vytvoření uzlu dokud není přítomen server. Čekání na odpověď od serveru, je pak implementována pomocí funkce `spin_until_future_complete()`

Action

Jedná se o rozšířenou verzi service. Akce z pravidla vykonává déle trvající požadavek. Například provedení řídícího manévru robota, který je prováděn v reálném světě a jeho provedení není tedy krátkodobá záležitost. Akce pak na rozdíl od service dokáže v průběhu vykonávání této činnosti odesílat průběžné aktualizace o aktuálním stavu provádění zpět volající node.

Implementačně funguje akce jako dva service a jeden topic. Cílový (goal) service slouží k zaslání požadavku na server a jeho potvrzení. Výsledkový (result) pak vrací výsledek operace. V průběhu akce pak server posílá aktualizace do topicu.

Algoritmus 6: Action server

```
1: self.action_server = ActionServer(self, Interface, "action_name",
   self.callback_function)
2: def callback_function(self, goal_handle):
3:     goal_handle.request.item
       // odeslání zpětné vazby volajícímu
4:     feedback = Interface.Feedback()
5:     feedback.item = some_value
6:     goal_handle.publish_feedback(feedback)
       // úspěšné ukončení požadavku
7:     goal_handle.succeed()
8:     result = Interface.Result()
9:     result.item = some_value
10:    return result
```

Action server využívá stejné postupy jako předchozí ukázky, pouze jich kombinuje více dohromady. V rámci callback funkce může odesílat průběžně zpětnou vazbu a nakonec jako návrat z funkce předá výsledek.

Při pohledu na implementaci action klienta lze dobře vidět vnitřní implementace akcí. První funkce `send_goal()` vypadá podobně jako service client. Dojde zde k zaslání požadavku na server a následné čekání na odpověď. Součástí požadavku je předání feedback topicu a čekání na odpověď není aktivní ale pomocí callback funkce. Druhá `response_callback_function` pak zpracuje výsledek požadavku (přijmutí nebo zamítnutí) a pošle požadavek na result service, který nakonec vrátí výsledek.

Interface

Interface slouží k určení přesného formátu jednotlivých zpráv, které jsou posílány mezi uzly. ROS2 obsahuje mnoho již vytvořených a vývojáři po celém světě používaných formátů. Tento přístup podporuje znovupoužitelnost vytvořeného kódu a šetří práci. Díky tomu může být software pro ovládání konkrétního kusu hardware naimplementován pouze jednou s využitím standartního rozhraní a všichni ostatní jej pak mohou využít ve svých systémech. Pokud však standartní interface nevyhovuje potřebám, lze si naimplementovat vlastní. K definici konkrétního formátu slouži tři druhy souborů.

Prvním jsou `.msg` zprávy. Tento formát je využívám topicy. Skládá se ze seznamu dvojic datový typ a název (případně ještě komentář).

```
int32 angle #comment
string direction
```

Druhým je `.srv`. Slouží pro definici request/response zpráv pro komunikaci se servicem. Tento soubor obsahuje dvě části, požadavek a odpověď, každá je tvořena seznamem položek a jsou odděleny řádkem `---`.

Algoritmus 7: Action client

```
self.action_client = ActionClient(self, Interface, "action_name")

// zaslaní požadavku na server
def send_goal(self):
    goal_msg = Servo.Goal()
    goal_msg.item = some_value
    self.action_client.wait_for_server()
    self.goal_future = self.action_client.send_goal_async(goal_msg,
    self.feedback_callback_function)
    self.goal_future.add_done_callback(self.response_callback_function)

// reakce na přijmutí nebo zamítnutí požadavku
def response_callback_function(self, future):
    goal_handle = future.result()
    if not goal_handle.accepted:
        return
    self.result_future = goal_handle.get_result_async()
    self.result_future.add_done_callback(self.result_callback_function)

def feedback_callback_function(self, msg):
    feedback = msg.feedback
    value = feedback.item
def result_callback_function(self, future):
    result = future.result().result
    value = result.item
```

```
int32 a
int32 b
---
int64 sum
```

Poslední je `.action` soubor. Slouží pro komunikaci s action serverem. Definice se skládá ze tří seznamů, jeden pro požadavek, druhý pro odpověď a poslední pro stavové aktualizace.

```
float32 goal_angle
---
bool response
---
float32 current_angle
```

Parametry

ROS2 uzly lze spouštět s parametry. Typicky slouží k nastavení hodnot (předání konfiguračního souboru) za běhu programu bez potřeby zásahu do zdrojových kódů. Příkladem může uzel, sloužící k obsluze periferního zařízení a parametrem jsou mu předány čísla GPIO pinů na které je dané zařízení připojeno.

Algoritmus 8: PARAMETERS

```
  // deklarace parametru, typicky v konstruktoru
1: self.declare_parameter('parameter_name', 'default_parameter_value')

  // získání hodnoty parametru
2: param =
  self.get_parameter('parameter_name').get_parameter_value().string_value


  // nastavení hodnoty parametru
3: new_param = rclpy.parameter.Parameter(
4:     'parameter_name',
5:     rclpy.Parameter.Type.STRING,
6:     'default_parameter_value'
7: )
8: new_param_list = [new_param]
9: self.set_parameters(new_param_list)
```

Launch File

ROS2 systém se skládá z velkého množství navzájem komunikujících uzlů, a protože spouštění každého uzlu zvlášť by bylo pracné a zdlouhavé, existují launch soubory, které tuto práci usnadňují. Tyto soubory můžou být napsány v pythonu, yaml nebo xml. Čtyři hlavní úkoly které launch soubory plní jsou, spouštění uzlu, volání dalšího launch soubory, nastavení parametrů a proměnných prostředí. Prakticky se launch soubory píší minimálně na dvou úrovních. Na nižší úrovni se využívají jako součást package, kde slouží k spouštění jednotlivých uzlů. Jejich úkolem je nastavit uzel tak, aby nebylo potřeba modifikovat zdrojový kód. Typicky tak nastavuje správný namespace, předávají parametry, konfigurační soubory a případně dochází k přemapování jména topicu. Launch soubory vyšších úrovní pak slouží ke spuštění několika uzlů zároveň a využívají k tomu launch soubory nacházející se v packages.

Knihovny

Algoritmus 9: Launch file

```
def generate_launch_description():
    return LaunchDescription([
        // spuštění konkrétního uzlu
        Node(
            package='package_name',
            executable='node_name',
            namespace='namespace_name',
            parameters=[{
                'param_name' : param_value,
            }]
        ),
        Node(
            package='package_name',
            executable='node_name',
            remappings=[
                ('topic_name', 'different_topic_name'),
            ]
        ),
        // zavolání dalšího launch souboru
        IncludeLaunchDescription(
            PythonLaunchDescriptionSource([
                PathJoinSubstitution([
                FindPackageShare('package_name'),
                    'launch',
                    'node_name.py'
                ])
            ]),
        )
    ])
```

Kapitola 3

Implementace

3.1 Struktura kódu

3.2 Manuální řízení

3.3 Sledování čáry

Kapitola 4

Závěr

Literatura

[1] *I2C-bus specification and user manual* [online]. um10204. NXP Semiconductors [cit. 2023-11-12]. Dostupné z: https://www.nxp.com/docs/en/user-guide/UM10204.pdf.

[2] NXP Semiconductors. *PCA9685: 16-channel, 12-bit PWM Fm+ I2C-bus LED controller*. 2009 [cit. 2023-11-12].

[3] Pyšný, R. *BiBTeX styl pro ČSN ISO 690 a ČSN ISO 690-2*. Brno, CZ, 2009. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Dostupné z: https://www.fit.vut.cz/study/thesis/7848/.

[4] Rico, F. M. *A concice introduction to robot programming with ROS2*. First edition. CRC Press, 2023. ISBN 978-1-032-26465-3.

[5] Rábová, Z., Hanáček, P., Peringer, P., Přikryl, P. a Křena, B. *Užitečné rady pro psaní práce* [online]. FIT VUT v Brně, listopad 2008 [cit. 2019-10-02]. Dostupné z: https://www.fit.vut.cz/study/theses/theses-advice/.

[6] Salhuana, M. *Sensor I2C Setup and FAQ* [online]. an4481. Freescale Semiconductor, Inc, červenec 2012 [cit. 2023-11-12]. Dostupné z: https://www.nxp.com/docs/en/application-note/AN4481.pdf.

[7] STMicroelectronics. *L298: Dual Full-Bridge Driver*. 2000 [cit. 2023-11-12].

Příloha A

Jak pracovat s touto šablonou

V této příloze je uveden popis jednotlivých částí šablony, po kterém následuje stručný návod, jak s touto šablonou pracovat. Pokud po jejím přečtení k šabloně budete mít nějaké dotazy, připomínky apod., neváhejte a napište na e-mail `sablona@fit.vutbr.cz`.

Popis částí šablony

Po rozbalení šablony naleznete následující soubory a adresáře:

bib-styles Styly literatury (viz níže).

obrazky-figures Adresář pro Vaše obrázky. Nyní obsahuje `placeholder.pdf` (tzv. TODO obrázek, který lze použít jako pomůcku při tvorbě technické zprávy), který se s prací neodevzdává. Název adresáře je vhodné zkrátit, aby byl jen ve zvoleném jazyce.

template-fig Obrázky šablony (znak VUT).

fitthesis.cls Šablona (definice vzhledu).

Makefile Makefile pro překlad, počítání normostran, sbalení apod. (viz níže).

projekt-01-kapitoly-chapters.tex Soubor pro Váš text (obsah nahraďte).

projekt-20-literatura-bibliography.bib Seznam literatury (viz níže).

projekt-30-prilohy-appendices.tex Soubor pro přílohy (obsah nahraďte).

projekt.tex Hlavní soubor práce – definice formálních částí.

Styl literatury v šabloně je od Ing. Radka Pyšného [3], jehož práce byla vylepšena prof. Adamem Heroutem, dr. Jaroslavem Dytrychem a panem Karlem Hanákem tak, aby odpovídala normě a podporovala všechny často využívané typy citací. Jeho dokumentaci naleznete v příloze E.

Makefile kromě překladu do PDF nabízí i další funkce:

- přejmenování souborů (viz níže),
- počítání normostran,
- spuštění vlny pro doplnění nezlomitelných mezer,
- sbalení výsledku pro odeslání vedoucímu ke kontrole (zkontrolujte, zda sbalí všechny Vámi přidané soubory, a případně doplňte).

Nezapomeňte, že vlna neřeší všechny nezlomitelné mezery. Vždy je třeba manuální kontrola, zda na konci řádku nezůstalo něco nevhodného – viz Internetová jazyková příručka[1].

Pozor na číslování stránek! Pokud má obsah 2 strany a na 2. jsou jen „Přílohy“ a „Seznam příloh“ (ale žádná příloha tam není), z nějakého důvodu se posune číslování stránek o 1 (obsah „nesedí“). Stejný efekt má, když je na 2. či 3. stránce obsahu jen „Literatura“ a je možné, že tohoto problému lze dosáhnout i jinak. Řešení je několik (od úpravy obsahu, přes nastavení počítadla až po sofistikovanější metody). **Před odevzdáním proto vždy překontrolujte číslování stran!**

Doporučený postup práce se šablonou

1. **Zkontrolujte, zda máte aktuální verzi šablony.** Máte-li šablonu z předchozího roku, na stránkách fakulty již může být novější verze šablony s aktualizovanými informacemi, opravenými chybami apod.
2. **Zvolte si jazyk**, ve kterém budete psát svoji technickou zprávu (česky, slovensky nebo anglicky) a svoji volbu konzultujte s vedoucím práce (nebyla-li dohodnuta předem). Pokud Vámi zvoleným jazykem technické zprávy není čeština, nastavte příslušný parametr šablony v souboru projekt.tex (např.: `documentclass[english]{fitthesis}` a přeložte prohlášení a poděkování do angličtiny či slovenštiny.
3. **Přejmenujte soubory.** Po rozbalení je v šabloně soubor `projekt.tex`. Pokud jej přeložíte, vznikne PDF s technickou zprávou pojmenované `projekt.pdf`. Když vedoucímu více studentů pošle `projekt.pdf` ke kontrole, musí je pracně přejmenovávat. Proto je vždy vhodné tento soubor přejmenovat tak, aby obsahoval Váš login a (případně zkrácené) téma práce. Vyhněte se však použití mezer, diakritiky a speciálních znaků. Vhodný název může být např.: „`xlogin00-Cisteni-a-extrakce-textu.tex`“. K přejmenování můžete využít i přiložený Makefile:

   ```
   make rename NAME=xlogin00-Cisteni-a-extrakce-textu
   ```

4. Vyplňte požadované položky v souboru, který byl původně pojmenován `projekt.tex`, tedy typ, rok (odevzdání), název práce, svoje jméno, ústav (dle zadání), tituly a jméno vedoucího, abstrakt, klíčová slova a další formální náležitosti.
5. Nahraďte obsah souborů s kapitolami práce, literaturou a přílohami obsahem svojí technické zprávy. Jednotlivé přílohy či kapitoly práce může být výhodné uložit do samostatných souborů – rozhodnete-li se pro toto řešení, je doporučeno zachovat konvenci pro názvy souborů, přičemž za číslem bude následovat název kapitoly.

[1] Internetová jazyková příručka http://prirucka.ujc.cas.cz/?id=880

6. Nepotřebujete-li přílohy, zakomentujte příslušnou část v `projekt.tex` a příslušný soubor vyprázdněte či smažte. Nesnažte se prosím vymyslet nějakou neúčelnou přílohu jen proto, aby daný soubor bylo čím naplnit. Vhodnou přílohou může být obsah přiloženého paměťového média.

7. Smažte soubory s kapitolami a přílohami pro jazyk, který jste nevyužili (s nebo bez `-en`).

8. Zadání, které si stáhnete v PDF z IS VUT (odkaz „Zadání pro vložení do práce“ či „Thesis assignment“), uložte do souboru `zadani.pdf` a povolte jeho vložení do práce parametrem šablony v `projekt.tex` (`documentclass[zadani]{fitthesis}`).

9. Nechcete-li odkazy tisknout barevně (bez konzultace s vedoucím příliš nedoporučuji), budete pro tisk vytvářet druhé PDF s tím, že nastavíte parametr šablony pro tisk: (`documentclass[zadani,print]{fitthesis}`). Budete-li tisknout barevně, místo `print` použijte parametr `cprint`. Barevné logo se nesmí tisknout černobíle!

10. Vzor desek, do kterých bude práce vyvázána, si vygenerujte v informačním systému fakulty u zadání. Pro disertační práci lze zapnout parametrem v šabloně `cover` (více naleznete v souboru `fitthesis.cls`).

11. Nezapomeňte, že zdrojové soubory i (obě verze) PDF musíte odevzdat na CD či jiném médiu přiloženém k technické zprávě.

Obsah práce se generuje standardním příkazem `\tableofcontents` (zahrnut v šabloně). Přílohy jsou v něm uvedeny úmyslně.

Pokyny pro oboustranný tisk

- **Oboustranný tisk je doporučeno konzultovat s vedoucím práce.**
- Je-li práce tištěna oboustranně a její tloušťka je menší než tloušťka desek, nevypadá to dobře.
- Zapíná se parametrem šablony: `\documentclass[twoside]{fitthesis}`
- Po vytištění oboustranného listu zkontrolujte, zda je při prosvícení sazební obrazec na obou stranách na stejné pozici. Méně kvalitní tiskárny s duplexní jednotkou mají často posun o 1–3 mm. Toto může být u některých tiskáren řešitelné tak, že vytisknete nejprve liché stránky, pak je dáte do stejného zásobníku a vytisknete sudé.
- Za titulním listem, obsahem, literaturou, úvodním listem příloh, seznamem příloh a případnými dalšími seznamy je třeba nechat volnou stránku, aby následující část začínala na liché stránce (`\cleardoublepage`).
- Konečný výsledek je nutné pečlivě překontrolovat.

Styl odstavců

Odstavce se zarovnávají do bloku a pro jejich formátování existuje více metod. U papírové literatury je častá metoda s použitím odstavcové zarážky, kdy se u jednotlivých odstavců textu odsazuje první řádek odstavce asi o jeden až dva čtverčíky, tedy přibližně o dvě šířky

velkého písmene M základního textu (vždy o stejnou, předem zvolenou hodnotu). Poslední řádek předchozího odstavce a první řádek následujícího odstavce se v takovém případě neoddělují svislou mezerou. Proklad mezi těmito řádky je stejný jako proklad mezi řádky uvnitř odstavce [5].

Další metodou je odsazení odstavců, které je časté u elektronické sazby textů. První řádek odstavce se při této metodě neodsazuje a mezi odstavce se vkládá vertikální mezera o velikosti 1/2 řádku. Obě metody lze v kvalifikační práci použít, nicméně často je vhodnější druhá z uvedených metod. Metody není vhodné kombinovat.

Jeden z výše uvedených způsobů je v šabloně nastaven jako výchozí, druhý můžete zvolit parametrem šablony „`odsaz`“.

Užitečné nástroje

Následující seznam není výčtem všech využitelných nástrojů. Máte-li vyzkoušený osvědčený nástroj, neváhejte jej využít. Pokud však nevíte, který nástroj si zvolit, můžete zvážit některý z následujících:

MikTeX LaTeX pro Windows – distribuce s jednoduchou instalací a vynikající automatizací stahování balíčků. MikTex obsahuje i vlastní editor, ale spíše doporučuji TeXstudio.

TeXstudio Přenositelné GUI pro LaTeX s otevřeným zdrojovým kódem (opensource). Ctrl+klik umožňuje přepínat mezi zdrojovým textem a PDF. Má integrovanou kontrolu pravopisu[2], zvýraznění syntaxe apod. Pro jeho využití je nejprve potřeba nainstalovat MikTeX, případně jinou LaTeXovou distribuci.

WinEdt Ve Windows je dobrá kombinace WinEdt + MiKTeX. WinEdt je GUI pro Windows, pro jehož využití je nejprve potřeba nainstalovat MikTeX či TeX Live.

Kile Editor pro desktopové prostředí KDE (Linux). Umožňuje živé zobrazení náhledu. Pro jeho využití je potřeba mít nainstalovaný TeX Live a Okular.

JabRef Pěkný a jednoduchý program v Javě pro správu souborů s bibliografií (literaturou). Není potřeba se nic učit – poskytuje jednoduché okno a formulář pro editaci položek.

InkScape Přenositelný opensource editor vektorové grafiky (SVG i PDF). Vynikající nástroj pro tvorbu obrázků do odborného textu. Jeho ovládnutí je obtížnější, ale výsledky stojí za to.

GIT Vynikající pro týmovou spolupráci na projektech, ale může výrazně pomoci i jednomu autorovi. Umožňuje jednoduché verzování, zálohování a přenášení mezi více počítači.

Overleaf Online nástroj pro LaTeX. Přímo zobrazuje náhled a umožňuje jednoduchou spolupráci (vedoucí může průběžně sledovat psaní práce), vyhledávání ve zdrojovém textu či ve vygenerovaném PDF, kontrolu pravopisu apod. Zdarma jej však lze využít pouze s určitými omezeními (někomu stačí na disertaci, jiný na ně může narazit i při psaní bakalářské práce) a pro dlouhé texty je pomalejší. FIT VUT v Brně má pro studenty i zaměstnance licenci, kterou si lze aktivovat na `https://www.overleaf.com/edu/but`.

[2] Českou kontrolu pravopisu lze doinstalovat z `https://extensions.openoffice.org/de/project/czech-dictionary-pack-ceske-slovniky-cs-cz`

Pozn.: Overleaf nepoužívá Makefile v šabloně – aby překlad fungoval, je v menu nutné zvolit `projekt.tex` jako hlavní dokument.

Příloha B

Psaní anglického textu

Tato příloha je převzata ze stránek doc. Černockého [**?**].

Spousta lidí píše zprávy k projektům anglicky (a to je dobře!), ale dělá v nich spoustu zbytečných chyb (a to je špatně). Nejsem angličtinář, ale tento jazyk už nějakých pár let používám k psaní, čtení i komunikaci – tato příloha obsahuje pár důležitých věcí. Pokud chcete napsat práci nebo článek opravdu 100 % dobře, nezbude Vám než si najmout rodilého mluvčího (a to by měl by být trochu technicky zdatný a aspoň trochu rozumět tomu, co píšete, ať to neskončí ještě hůř . . .).

Obecně

- Předtím, než budete sami něco psát, si přečtěte pár anglických technických článků a zkuste si zapamatovat a získat „obecný pocit“, jak se to píše.
- Používejte vždy korektor pravopisu – zabudovaný ve Wordu, nebo v OpenOffice, pokud děláte na Linuxu, tak ISPELL a další (většina editorů pro LaTeX má již kontrolu pravopisu integrovanou).
- Používejte korektor gramatiky. Nevím, jestli je nějaký dostupný na Linuxu, ale ten ve Wordu celkem slušně funguje a pokud Vám něco zelené podtrhne, je tam většinou opravdu chyba. Můžete do něj nakopírovat i zdrojový text pro LaTeX, opravit, a pak uložit opět jako čistý text. Pokud používáte vim, je tam zabudovaný také a zvládne jak překlepy, tak základní gramatiku. V dokumentu `diplomka.tex` na první řádek napište:

  ```
  % vim:spelllang=en_us:spell
  ```

  (případně `en_gb` pro OED angličtinu) *Poznámka editora:* Existuje i velmi dobrý online nástroj Grammarly[1], který je v základní verzi zdarma.
- Online slovníky jsou dobré, ale nepoužívejte je slepě. Většinou dají více variant a ne každá je správně.

[1] https://www.grammarly.com/

- Na vyhledávání a zjištění, co bude asi správné, můžete použít Google. Např.: nevíte, jak se řekne „výhoda tohoto přístupu“. Slovník na seznam.cz dá asi 10 variant. Napište je postupně do vyhledávání na googlu:

  ```
  "advantage of this approach" 1100000 hits
  "privilege of this approach" 6 hits
  "facility of this approach"  16 hits
  ```

  Neříkám, že je to 100 % správně, ale je to určité vodítko. Toto se dá použít i na dohledání správných spojek (třeba „among two cases“ nebo „between two cases“?)

SVOMPT a shoda

Struktura anglické věty je SVOPMT: SUBJECT VERB OBJECT MANNER PLACE TIME a přes to nejede vlak! Není volná jako v češtině. Jinak to je maximálně v nějaké divadelní hře, kde je potřeba něco zdůraznit. Hlavně podmět tam musí vždy být, na to se často zapomíná, protože v CZ/SK může být zamlčený nebo nevyjádřený. SVOMPT platí i ve vedlejších větách!

```
BAD: We have shown that is faster than the other function.
GOOD: We have shown that it is faster than the other function.
```

Shoda podmětu s přísudkem – zní to šíleně, ale dělá se v tom spousta chyb.

```
he has
the users have
people were
```

Členy

Členy v angličtině jsou noční můra a téměř nikdo z nás je nedává dobře. Základní pravidlo je, že když je něco určitého, musí předtím být „the“. Členy musí být určitě u těchto spojení:

```
the first, the second, ...
the last
the most (třetí stupeň přídavných jmen a príslovcí) ...
the whole
the following
the figure, the table.
the left, the right - on the left pannel, from the left to the right ...
```

Naopak člen NESMÍ být, pokud používáte přesné označení obrázku, kapitoly atd.

```
in Figure 3.2
in Chapter 7
in Table 6.4
```

Pozor na „a“ vs. „an“, řídí se to podle výslovnosti a ne podle toho, jak je slovo napsané, takže:

```
an HMM
an XML
a universal model
a user
```

Slovesa

Pozor na trpné tvary sloves – u pravidelných je to většinou bez problémů, u nepravidelných často špatně, typicky

```
packet was sent (ne send)
approach was chosen (ne choosed)
```

... vetšinou to opraví korektor pravopisu, ale někdy ne.

Pozor na časy, občas je v nich pěkný nepořádek. Pokud něco nějak obecně je, přítomný čas. Pokud jste něco udělali, minulý. Pokud to dalo nějaký výsledek a ten výsledek teď existuje a třeba ho nějak diskutujete, přítomný. Nepoužívejte příliš složité časy jako je předpřítomný a vůbec ne předminulý pokud nevíte přesně, co děláte.

```
JFA is a technique that works for everyone in speaker recognition.
We implemented it according to Kenny's recipe in \cite{Kenny}.
12000 segments from NIST SRE 2006 were processed. When compared
with a GMM baseline, the results are completely bad.
```

Délka vět a struktura

- Pište kratší věty a souvětí, pokud máte něco na 5 řádků, většinou se to nedá číst.
- Strukturujte věty pomocí čárek (více než v češtině!), hlavně po úvodu věty, po kterém začíná vlastní věta. Někdy se dává čárka i před „and“ (na rozdíl od češtiny).

```
In this chapter, we will investigate ...
The first technique did not work, the second did not work as well,
and the third one also did not work.
```

Specifika technického textu

Píšete technický text, proto nepoužívejte zkratky

```
he's
gonna
Petr's working on ...
```

a podobně. Jediné, které je tolerované, je „doesn't“, ale neuděláte chybu, když napíšete „does not“.

V technických textech se spíš používá trpný rod než činný:

```
BAD: In this chapter, I describe used programming languages.
GOOD: In this chapter, used programming languages are described.
```

Pokud už činný použijete, dává se v technických textech spíše „we“, i když na práci děláte sami. „I“, „my“ atd. se používají pouze tam, kde jde o to zdůraznit, že jde o Vaši osobu, tedy třeba v závěru nebo v popisu „original claims“ v disertaci.

Časté chyby ve slovech

- Pozor na jeho/její, není to it's, ale its.
- Obrázek není picture, ale figure.
- Spojka „než“ je „than“, ne „then“ – bigger than this, smaller than this ... hrozně častá chyba! „Then“ je pak, potom.

Příloha C

Checklist

Tento checklist byl převzat ze šablony pro kvalifikační práce, která je k dispozici na blogu prof. Herouta [**?**], který s laskavým dovolením využil nápadu dr. Szökeho[1].

Velká bezpečnost letecké dopravy stojí z části na tom, že lidé kolem letadel mají **checklisty** na úplně každý, třeba rutinní a dobře zažitý, postup. Jako pilot strpí to, že bude trochu za blbce a opravdu tužtičkou do seznamu úkonů odškrtá dokonale zvládnuté akce, vytiskněte si a odškrtejte před odevzdáním diplomky i vy tento checklist a vyhněte se tak častým chybám, které by mohly mít až fatální následky na výsledné hodnocení Vaší práce.

Struktura

- ☐ Už ze samotných názvů a struktury kapitol je patrné, že bylo splněno zadání.
- ☐ V textu se nevyskytuje kapitola, která by měla méně než čtyři strany (kromě úvodu a závěru). Pokud ano, radil(a) jsem se o tom s vedoucím a ten to schválil.

Obrázky a grafy

- ☐ Všechny obrázky a tabulky byly zkontrolovány a jsou poblíž místa, odkud jsou z textu odkazovány, takže nebude problém je najít.
- ☐ Všechny obrázky a tabulky mají takový popisek, že celý obrázek dává smysl sám o sobě, bez čtení dalšího textu. Vůbec nevadí, když má popisek několik řádků.
- ☐ Pokud je obrázek převzatý, tak je to v popisku zmíněno: „Převzato z [X].“
- ☐ Písmenka ve všech obrázcích používají font podobné velikosti, jako je okolní text (ani výrazně větší, ani výrazně menší).
- ☐ Grafy a schémata jsou vektorově (tj. v PDF).
- ☐ Snímky obrazovky nepoužívají ztrátovou kompresi (jsou v PNG).
- ☐ Všechny obrázky jsou odkázány z textu.
- ☐ Grafy mají popsané osy (název osy, jednotky, hodnoty) a podle potřeby mřížku.

[1] http://blog.igor.szoke.cz/2017/04/predstartovni-priprava-letu-neni.html

Rovnice

- ☐ Identifikátory a jejich indexy v rovnicích jsou jednopísmenné (kromě nečastých zvláštních případů jako $t_{\max}$).
- ☐ Rovnice jsou číslovány.
- ☐ Za (nebo vzácně před) rovnicí jsou vysvětleny všechny proměnné a funkce, které zatím vysvětleny nebyly.

Citace

- ☐ **Všechny použité zdroje jsou citovány.**
- ☐ Adresy URL odkazující na služby, projekty, zdroje, github apod. jsou odkazovány pomocí `\footnote{\url{...}}`.
- ☐ Všechny citace používají správné typy.
- ☐ Citace mají autora, název, vydavatele (název konference), rok vydání. Když některá nemá, je to dobře zdůvodněný zvláštní případ a vedoucí to odsouhlasil.
- ☐ Je-li ve zdrojových textech programu něco převzaté, je to tam řádně citováno v souladu s licencí.
- ☐ Je-li podstatná část zdrojových textů programu převzatá, je toto zmíněno v textu práce a je citován zdroj.

Typografie

- ☐ Žádný řádek nepřetéká přes pravý okraj.
- ☐ Na konci řádku nikde není jednopísmenná předložka (spraví to nedělitelná mezera ~).
- ☐ Číslo obrázku, tabulky, rovnice, citace není nikde první na novém řádku (spraví to nedělitelná mezera ~).
- ☐ Před číselným odkazem na poznámku pod čarou nikde není mezera (to jest vždy takto[2], nikoliv takto [3]).

Jazyk

- ☐ Použil jsem kontrolu pravopisu a v textu nikde nejsou překlepy.
- ☐ Nechal jsem si text přečíst od (alespoň) jednoho dalšího člověka, který umí dobře česky / anglicky / slovensky.
- ☐ V práci psané česky nebo slovensky abstrakt zkontroloval někdo, kdo umí opravdu dobře anglicky.
- ☐ V textu se nikde nepoužívá druhá mluvnická osoba (vy/ty).

[2] příklad poznámky pod čarou
[3] jiný příklad poznámky pod čarou

☐ Když se v textu vyskytuje první mluvnická osoba (já, my), vždy se popisuje subjektivní záležitost (*rozhodl jsem se*, *navrhl jsem*, *zaměřil jsem se na*, *zjistil jsem* apod.).

☐ V textu se nikde nepoužívají hovorové výrazy.

☐ V českém či slovenském textu se zbytečně nepoužívají anglické výrazy, které mají ustálené české překlady. Např. slovo *defaultní* se nahradí např. slovem *implicitní* nebo *výchozí*.

Výsledek na datovém médiu, tj. software

☐ Mám připravené nepřepisovatelné datové médium

- CD-R,
- DVD-R,
- DVD+R ve formátu ISO9660 (s rozšířením RockRidge a/nebo Jolliet) nebo UDF,
- paměťová karta SD (Secure Digital) ve formátu FAT32 nebo exFAT s nastavenou ochranou proti přepisu.

☐ Pokud je výsledek online (služba, aplikace, ...), URL je viditelně v úvodu a závěru, aby bylo jasné, kde výsledek hledat.

☐ Na médiu nechybí povinné:

- zdrojové kódy (např. Matlab, C/C++, Python, ...)
- knihovny potřebné pro překlad,
- přeložené řešení,
- PDF s technickou zprávou (je-li pro tisk 2. verze, tak obě),
- zdrojový kód zprávy (LaTeX),

a případně volitelně po dohodě s vedoucím práce

- relevantní (např. testovací) data,
- demonstrační video,
- PDF plakátku,
- ...

☐ Zdrojové kódy jsou refaktorovány, komentovány a označeny hlavičkou s autorstvím, takže se v nich snadno vyzná i někdo další, než sám autor.

☐ Jakákoliv převzatá část zdrojového kódu je řádně citována – tedy označena úvodním a v případě převzetí více řádků i ukončovacím komentářem. Komentář obsahuje vše, co vyžaduje licence uvedená na webu (vždy je nutné se ji pokusit najít – např. Stack Overflow[4] má striktní pravidla pro citace).

[4] https://stackoverflow.blog/2009/06/25/attribution-required/

Odevzdání

- ☐ Chci práci (na max. 3 roky) utajit? Pokud ano, nejpozději měsíc před termínem odevzdání práce si podám žádost (v IS), ke které přiložím případné stanovisko firmy, jejíž duševní vlastnictví je třeba chránit.
- ☐ Mám splněný minimální počet normostran textu (lze spočítat pomocí Makefile a odhadem přičíst obrázky). Pokud jsem těsně pod minimem, konzultoval(a) jsem to s vedoucím.
- ☐ Pokud chci tisknout oboustranně, konzultoval(a) jsem to s vedoucím a mám správně nastavenou šablonu. Kapitoly začínají na liché stránce.
- ☐ Technickou zprávu mám v deskách z knihařství (min. 1 výtisk, při utajení oba).
- ☐ Za titulním listem práce je zadání (tzn. mám jej stažené z IS a vložené do šablony).
- ☐ V IS jsou abstrakty a klíčová slova.
    - V abstraktu a klíčových slovech v IS nejsou zkopírované vlnky pro nezlomitelné mezery.
- ☐ V IS je PDF práce (s klikatelnými odkazy).
- ☐ Oba výtisky práce jsou podepsané.
- ☐ V jednom (při utajení obou) výtisku práce je paměťové médium, na kterém je fixkou napsaný login (fixku na CD lze zapůjčit v knihovně, na Studijním oddělení nebo až při odevzdání).

Příloha D

LaTeXpro začátečníky

V této kapitole jsou uvedeny některé často využívané balíčky a příkazy pro LaTeX, které mohou být při tvorbě práce potřeba.

Užitečné balíčky

Studenti při sazbě textu často řeší stejné problémy. Některé z nich lze vyřešit následujícími balíčky pro LaTeX:

- `amsmath` – rozšířené možnosti sazby rovnic,
- `float, afterpage, placeins` – úprava umístění obrázků/tabulek (specifikátor `H`),
- `fancyvrb, alltt` – úpravy vlastností prostředí Verbatim,
- `makecell` – rozšíření možností tabulek,
- `pdflscape, rotating` – natočení stránky o 90 stupňů (pro obrázek či tabulku),
- `hyphenat` – úpravy dělení slov,
- `picture, epic, eepic` – přímé kreslení obrázků.

Některé balíčky jsou využity přímo v šabloně (v dolní části souboru `fitthesis.cls`). Nahlédnutí do jejich dokumentace může být rovněž velmi užitečné.

Sloupec tabulky zarovnaný vlevo s pevnou šířkou je v šabloně definovaný „L“ (používá se jako „p“).

Pro odkazování v rámci textu použijte příkaz `\ref{navesti}`. Podle umístění návěští se bude jednat o číslo kapitoly, podkapitoly, obrázku, tabulky nebo podobného číslovaného prvku). Pokud chcete odkázat stránku práce, použijte příkaz `pageref{navesti}`. Pro citaci literárního odkazu `\cite{identifikator}`. Pro odkazy na rovnice lze použít příkaz `\eqref{navesti}`.

Znak – (pomlčka) se V LaTeXu vkládá jako dvě mínus za sebou: --.

Často využívané příkazy pro LaTeX

Doporučuji nahlédnout do zdrojového textu této podkapitoly a podívat se, jak jsou následující ukázky vysázeny. Ve zdrojovém textu jsou i pomocné komentáře.

Příklad tabulky:

Tabulka D.1: Tabulka hodnocení

Jméno		
Jméno	Příjmení	Hodnocení
Jan	Novák	7.5
Petr	Novák	2

Příklad rovnice:

$$\cos^3\theta = \frac{1}{4}\cos\theta + \frac{3}{4}\cos 3\theta \tag{D.1}$$

a dvou horizontálně zarovnaných rovnic:

$$3x = 6y + 12 \tag{D.2}$$
$$x = 2y + 4 \tag{D.3}$$

Pokud je třeba rovnici citovat v textu, lze použít příkaz `\eqref`. Například na rovnici výše lze odkázat (D.1). Pokud chcete srovnat číslo rovnic u soustavy, lze použít prostředí `split`:

$$\begin{aligned} 3x &= 6y + 12 \\ x &= 2y + 4 \end{aligned} \tag{D.4}$$

Matematické symboly (α) a výrazy lze umístit i do textu $\cos\pi = -1$ a mohou být i v poznámce pod čarou[1].

Obrázek D.1 ukazuje široký obrázek složený z více menších obrázků. Klasický rastrový obrázek se vkládá tak, jak je vidět na obrázku D.2.

Obrázek D.1: **Široký obrázek.** Obrázek může být složen z více menších obrázků. Chcete-li se na tyto dílčí obrázky odkazovat z textu, využijte balíček `subcaption`.

Někdy je potřeba do příloh umístit diagram, který se nevejde na stránku formátu A4. Pak je možné vložit jednu stránku formátu A3 a do práce ji poskládat (tzv. skládání do Z,

[1] Vzorec v poznámce pod čarou: $\cos\pi = -1$

Obrázek D.2: Dobrý text je špatným textem, který byl několikrát přepsán. Nebojte se prostě něčím začít.

kdy se vytvoří dva sklady – lícem dolů a lícem nahoru, angl. Engineering fold – existuje i anglický pojem Z-fold, ale při tom by byl problém s vazbou). Přepnutí se provádí následovně: `\eject \pdfpagewidth=420mm` (pro přepnutí zpět pak 210mm).

Další často využívané příkazy naleznete ve zdrojovém textu ukázkového obsahu této šablony.

Příloha E

Příklady bibliografických citací

Styl czplain vychází ze stylu vytvořeného v rámci práce pana Pyšného [3]. Obsahuje sadu podporovaných typů citací s konkrétními příklady bibliografických citací.

Na následujících stránkách přílohy jsou uvedeny příklady, jenž znázorňují bibliografické citace následujících publikací a jejich částí:

- článku v seriálové publikaci (časopisu) (str. 38),
- monografické publikace (str. 39),
- sborníku (str. 40),
- článku ve sborníku nebo kapitoly v knize (str. 41),
- manuálu, dokumentace, technické zprávy a nepublikovaných materiálů (str. 42),
- akademické práce (str. 43),
- webové stránky (str. 44),
- a webové domény (str. 45).

Položky jsou označený barevně podle povinnosti:

- prvek je dle normy povinný
- prvek, který je dle normy volitelný
- prvek, který je dle normy povinný pro online informační zdroje
- prvek, který není předepsán normou a je v bibliografickém stylu v šabloně volitelný

Povinné položky se uvádí pouze pokud existují.

V souboru s bibliografií se záznamy uvádí následujícím způsobem:

```
@Article{Doe:2020,
   author               = "Doe, John",
   title                = "Jak citovat",
   subtitle             = "Citace článku",
   journal              = "Seriál o tvorbě prací",
   journalsubtitle      = "Formální náležitosti",
   howpublished         = "online",
   address              = "Brno",
   publisher            = "Fakulta informačních technologií VUT v Brně",
   contributory         = "Přeložil Jan NOVÁK",
   edition              = "1",
   version              = "verze 1.0",
   month                = 2,
   year                 = "2020",
   revised              = "revidováno 12. 2. 2020",
   volume               = "4",
   number               = "24",
   pages                = "8--21",
   cited                = "2020-02-12",
   doi                  = "10.1000/BC1.0",
   issn                 = "1234-5678",
   note                 = "Toto je zcela vymyšlená citace",
   url                  = "https://merlin.fit.vutbr.cz"
}
```

Citace jsou seřazeny podle abecedy. Řazení jmen s písmeny s diakritikou můžeme ovlivnit prvkem `key`, jehož hodnotu nastavíme na příjmení bez diakritiky. Pokud není vyplněn autor, citace se řadí na začátek seznamu, což není vhodné. Řazení v tomto případě můžeme taktéž ovlivnit vhodně nastaveným prvkem key.

Příklad:

```
   @Article{Cech:2020:Citace,
    author               = "Čech, Jan",
    key                  = "Cech",
    ...
```

Článek v seriálové publikaci – @Article

Položky záznamu

Prvek	Zápis v BibTeXu	Příklad
Tvůrce	author	Doe, John
Název příspěvku	title	Jak citovat
Vedlejší název	subtitle	Citace článku
Název seriálové publikace	journal	Seriál o tvorbě prací
Vedlejší názvy seriálu	journalsubtitle	Formální náležitosti
Typ nosiče	howpublished	online
Vydání	edition	1
Verze	version	verze 1.0
Další tvůrce	contributory	Přeložil Jan NOVÁK
Místo vydání	address	Brno
Nakladatel	publisher	Fakulta informačních technologií VUT v Brně
Měsíc	month	2
Rok	year	2020
Svazek	volume	4
Číslo	number	24
Rozsah příspěvku	pages	8-21
Revize	revised	revidováno 12. 2. 2020
Datum citování	cited	2020-02-12
Název edice	series	Návody k tvorbě prací
Číslo edice	editionnumber	42
Identifikátor digitálního obsahu	doi	10.1000/BC1.0
Standardní číslo	issn	1234-5678
Poznámky	note	Toto je zcela vymyšlená citace
Dostupnost a přístup	url	https://merlin.fit.vutbr.cz

Bibliografická citace

Doe, J. Jak Citovat: Citace článku. *Seriál o tvorbě prací: Formální náležitosti* [online]. 1. vyd., verze 1.0. Přeložil Jan NOVÁK. Brno: Fakulta informačních technologií VUT v Brně. Únor 2020, sv. 4, č. 24, s. 8–21, revidováno 12. 2. 2020, [cit. 2020-02-12]. Návody k tvorbě prací, č. 42. DOI: 10.1000/BC1.0. ISSN 1234-5678. Toto je zcela vymyšlená citace. Dostupné z: https://merlin.fit.vutbr.cz

Monografická publikace – @Book, @Booklet (kniha, brožura)

Položky záznamu

Prvek	Zápis v BibTeXu	Příklad
Tvůrce	author	John von Doe
Titul	title	Jak citovat
Vedlejší názvy	subtitle	Citace monografické publikace
Typ nosiče	howpublished	online
Vydání	edition	1
Další tvůrce	contributory	Přeložil Jan NOVÁK
Místo vydání	address	Brno
Nakladatel	publisher	Fakulta informačních technologií VUT v Brně
Měsíc vydání	month	2
Rok vydání	year	2020
Revize	revision	revidováno 12. 2. 2020
Datum citování	cited	2020-02-12
Rozsah	pages	220
Edice	series	Návody k tvorbě prací
Číslo edice	editionnumber	2
Standardní číslo	isbn	01-234-5678-9
Poznámky	note	Toto je zcela vymyšlená citace
Dostupnost a přístup	url	https://merlin.fit.vutbr.cz

Bibliografická citace

VON DOE, J. *Jak citovat: Citace monografické publikace* [online] . 1. vyd. Přeložil Jan NOVÁK. Brno:Fakulta informačních technologií VUT v Brně, únor 2020, revidováno 12. 2. 2020 [cit. 2020-02-12]. 220 s. Návody k tvorbě prací, č. 2. ISBN 01-234-5678-9. Toto je zcela vymyšlená citace. Dostupné z: `https://merlin.fit.vutbr.cz`

Sborník – @Proceedings

Položky záznamu

Prvek	Zápis v BibTeXu	Příklad
Tvůrce*	author	Čechmánek, Jan
Editor*	editor	Čechmánek, Jan
Titul	title	Jak citovat
Vedlejší názvy	subtitle	Citace monografické publikace
Typ nosiče	howpublished	online
Vydání	edition	1
Další tvůrce	contributory	Přeložil Jan NOVÁK
Místo vydání	address	Brno
Nakladatel	publisher	Fakulta informačních technologií VUT v Brně
Měsíc vydání	month	2
Rok vydání	year	2020
Svazek	volume	4
Číslo svazku	number	24
Rozsah příspěvku	pages	8-21
Revize	revised	revidováno 12. 2. 2020
Datum citování	cited	2020-02-12
Edice	series	Návody k tvorbě prací
Číslo edice	editionnumber	2
Identifikátor digitálního objektu	doi	10.1000/BC1.0
Standardní číslo	isbn nebo issn	01-234-5678-9
Poznámky	note	Toto je zcela vymyšlná citace
Dostupnost a přístup	url	https://merlin.fit.vutbr.cz

*Uvádí se buď autor, nebo editor.

Bibliografická citace

Čechmánek, J. *Jak citovat: Citace sborníku* [online]. 1. vyd. Přeložil Jan NOVÁK. Brno: Fakulta informačních technologií VUT v Brně, únor 2020, sv. 4, č. 24, s. 8–21, revidováno 12. 2. 2020 [cit. 2020-02-12]. Návody k tvorbě prací, č. 2. DOI: 10.1000/BC1.0. ISBN 01-234-5678-9. Toto je zcela vymyšlená citace. Dostupné z: https://merlin.fit.vutbr.cz

Článek ve sborníku nebo kapitola v knize – @InProceedings, @InCollection, @Conference, @InBook

Položky záznamu

Prvek	Zápis v BibTeXu	Příklad
Tvůrce	author	John von Doe
Název příspěvku	title	Jak citovat
Vedlejší názvy	subtitle	Citace článku
Jméno tvůrce mateřského dokumentu	editor nebo organisation	Smith, Peter
Název mateřského dokumentu	booktitle	Sborník konference o tvorbě prací
Vedlejší názvy mateřského dokumentu	booksubtitle	Formální náležitosti
Typ nosiče	howpublished	online
Vydání	edition	1
Verze	version	verze 1.0
Další původce mateřského dokumentu	contributory	Přeložil Jan NOVÁK
Místo vydání	address	Brno
Nakladatel	publisher	Fakulta informačních technologií VUT v Brně
Měsíc	month	2
Rok	year	2020
Svazek	volume	4
Číslo svazku	number	24
Kapitola	chapter	5
Rozsah příspěvku	pages	8-21
Revize	revised	revidováno 12. 2. 2020
Datum citování	cited	2020-02-12
Edice	series	Návody k tvorbě prací
Číslo edice	editionnumber	2
Standardní číslo	isbn nebo issn	1234-5678
Poznámky	note	Toto je zcela vymyšlená citace
Dostupnost a přístup	url	https://merlin.fit.vutbr.cz

Bibliografická citace

Doe, J. Jak citovat: Citace článku. In: Smith, P., ed. *Sborník konference o tvorbě prací: Formální náležitosti* [online]. 1. vyd., verze 1.0. Přeložil Jan NOVÁK. Brno: Fakulta informačních technologií VUT v Brně, únor 2020, sv. 4, č. 24, kap. 5, s. 8–21, revidováno 12. 2. 2020 [cit. 2020-02-12]. Návody k tvorbě prací, č. 2. ISSN 1234-5678. Toto je zcela vymyšlená citace. Dostupné z: https://merlin.fit.vutbr.cz

Manuál, dokumentace, technická zpráva a nepublikované materiály – @Manual, @TechReport, @Unpublished

Položky záznamu

Prvek	Zápis v BibTeXu	Příklad
Tvůrce (osoba nebo organizace)	author	Fakulta informačních technologií VUT v Brně
Titul	title	Manuál k tvorbě prací
Vedlejší názvy	subtitle	Citace manuálu
Typ nosiče	howpublished	online
Typ dokumantu	type	Uživatelský manuál
Číslo dokumentu	number	3
Vydání	edition	1
Další tvůrce	contributory	Editoval Jan NOVÁK
Místo vydání	address	Brno
Organizace nebo instituce	organization nebo institution	Fakulta informačních technologií VUT v Brně
Měsíc vydání	month	2
Rok vydání	year	2020
Revize	revised	revidováno 12. 2. 2020
Datum citování	cited	2020-02-12
Rozsah	pages	220
Poznámky	note	Toto je zcela vymyšlená citace
Dostupnost a přístup	url	https://merlin.fit.vutbr.cz

Bibliografická citace

Fakulta informačních technologií VUT v Brně. *Manuál k tvorbě prací: Citace manuálu* [online]. Uživatelský manuál 3, 1. vyd. Editoval Jan NOVÁK. Brno: Fakulta informačních technologií VUT v Brně, únor 2020, revidováno 12. 2. 2020 [cit. 2020-02-12]. 220 s. Toto je zcela vymyšlená citace. Dostupné z: `https://merlin.fit.vutbr.cz`

Akademická práce – @BachelorsThesis, @MastersThesis, @PhdThesis, @Thesis

Položky záznamu

Prvek	Zápis v BibTeXu	Příklad
Tvůrce	author	Fakulta informačních technologií VUT v Brně
Titul	title	BiBTeX styl pro ČSN ISO 690 a ČSN ISO 690-2
Vedlejší názvy	subtitle	
Typ nosiče	howpublished	online
Typ dokumantu	type	Diplomová práce
Místo vydání	address nebo location	Brno
Škola	school	Vysoké učení technické v Brně, Fakulta informačních technologií
Rok vydání	year	2020
Datum citování	cited	2020-02-12
Rozsah	pages	220
Rozsah příloh	inserts	20
Standardní číslo	isbn	01-234-5678-9
Vedoucí práce	Supervisor	Dytrych, Jaroslav
Poznámky	note	Toto je zcela vymyšlená citace
Dostupnost a přístup	url	https://www.fit.vut.cz/-study/theses

Bibliografická citace

Novák, J. *BiBTeX styl pro ČSN ISO 690 a ČSN ISO 690-2* [online]. Brno, CZ, 2020. [cit. 2020-02-12]. 80 s., 20. s. příl. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. ISBN 01-2345-678-9. Vedoucí práce Dytrych, J. Toto je zcela vymyšlená citace. Dostupné z: `https://www.fit.vut.cz/study/theses`

Webová stránka – @Webpage

Položky záznamu

Prvek	Zápis v BibTeXu	Příklad
Tvůrce	author	Nováková, Jana
Název příspěvku	secondarytitle	Citace příspěvku
Název stránky	title	Web tvorby prací
Vedlejší název stránky	subtitle	
Typ nosiče	howpublished	online
Další tvůrce	contributory	Editoval Jan NOVÁK
Verze	version	Verze 1.0
Místo vydání	address	Brno
Vydavatel	publisher	Fakulta informačních technologií VUT v Brně
Den	day	12
Měsíc vydání	month	2
Rok vydání	year	2020
Čas publikování	time	14:00
Revize	revised	Revidováno 12. 2. 2020
Identifikátor digitálního objektu	doi	10.1000/BC1.0
Standardní číslo	issn	1234-5678
Poznámky	note	Toto je zcela vymyšlená citace
Dostupnost a přístup	url	https://merlin.fit.vutbr.cz
Cesta	path	Domů; Umění; Umění citace

Bibliografická citace

Nováková, J. Citace příspěvku. *Web tvorby prací* [online]. Editoval Jan NOVÁK. Verze 1.0. Brno: Fakulta informačních technologií VUT v Brně, 2. března 1998 14:10. Revidováno 12. 2. 2020 [cit. 2020-02-12]. DOI: 10.1000/BC1.0. ISSN 1234-5678. Toto je zcela vymyšlená citace. Dostupné z: https://merlin.fit.vutbr.cz Path: Domů; Umění; Umění citace.

Webová doména – @Website

Položky záznamu

Prvek	Zápis v BibTeXu	Příklad
Tvůrce (osoba nebo organizace)	author	Nováková, Jana
Název webu	title	Web tvorby prací
Vedlejší název webu	subtitle	
Typ nosiče	howpublished	online
Další tvůrce	contributory	Editoval Jan NOVÁK
Verze	version	Verze 1.0
Místo vydání	address	Brno
Vydavatel	publisher	Fakulta informačních technologií VUT v Brně
Den	day	12
Měsíc vydání	month	2
Rok vydání	year	2020
Čas publikování	time	14:00
Revize	revised	Revidováno 12. 2. 2020
Datum	citování	cited 2020-02-12
Identifikátor digitálního objektu	doi	10.1000/BC1.0
Standardní číslo	issn	1234-5678
Poznámky	note	Toto je zcela vymyšlená citace
Dostupnost a přístup	url	https://merlin.fit.vutbr.cz

Bibliografická citace

Nováková, J. *Web tvorby prací* [online]. Editoval Jan NOVÁK. Verze 1.0. Brno: Fakulta informačních technologií VUT v Brně, 2. března 1998 14:10. Revidováno 12. 2. 2020 [cit. 2020-02-12]. DOI: 10.1000/BC1.0. ISSN 1234-5678. Toto je zcela vymyšlená citace. Dostupné z: https://merlin.fit.vutbr.cz.