



In-Time Server

Manuál v1.6

Paul Gingerbread



In-Time Server – Manuál v1.6

Autor: Paul Gingerbread

Sazba a tisk: Systémové nakladatelství TimePress, Lidechko 2025

Náklad: 25 výtisků, první vydání

ISBN: 978-80-420-6900-0

Obsah

| | |
|--|----------|
| 1 Úvod | 1 |
| Úvod | 1 |
| 2 Uvedení do provozu | 1 |
| 2.1 Zapojení | 1 |
| 2.2 Start | 2 |
| 2.2.1 Fáze BIOS/UEFI (po zapnutí) | 2 |
| 2.2.2 Zavaděč GRUB | 2 |
| 2.2.3 Načítání jádra a <i>initramfs</i> | 2 |
| 2.2.4 systemd : spouštění služeb | 3 |
| 2.2.5 Přihlašovací výzva (TTY1) | 3 |
| 3 Prostředí příkazové řádky | 1 |
| 3.1 Co je <i>shell</i> a jak do něj psát | 1 |
| 3.2 Základní editace a ovládání klávesnicí | 1 |
| 3.3 Co je příkaz, argument a přepínač | 2 |
| 3.4 Tipy a triky | 2 |
| 4 Použití příkazu time_server | 1 |
| 4.1 Obecné přepínače | 1 |
| 4.2 Modifikační přepínače | 1 |
| 4.3 Kritické operace | 2 |
| 4.4 Režimy a autentizace | 2 |
| 5 Příklad použití | 3 |
| 5.1 Zjistit celkový čas v systému | 3 |
| 5.2 Výpis zbývajících časů uživatelů | 3 |
| 5.3 Výpis registru uživatelů | 4 |
| 5.4 Přidat/odebrat čas uživateli | 4 |
| 5.5 Vstup do chráněného režimu | 4 |
| 5.6 Spustit systém | 5 |

Kapitola 1

Úvod

In-Time Server Admin je příkazová utilita pro správu a řízení běžícího časového serveru systému *In-Time*. Tento server vyměřuje čas občanům Lidové časové federace a představuje klíčovou infrastrukturu, bez které by nemohly fungovat základní společenské procesy. Nástroj je určen výhradně pro členy *časového výboru*, který nad systémem drží dohled, řídí jeho provoz a nese za něj plnou odpovědnost. Inspirací je filmový svět *In Time*, v němž čas není jen údaj na displeji, ale primární měna a zároveň délka života. Stejná logika zde platí i prakticky: jakákoliv změna v záznamech času se bezprostředně promítá do životů lidí.

Tento manuál poskytuje srozumitelný, ale detailní popis funkcí nástroje, jeho konfiguračních přepínačů a doporučených pracovních postupů. Cílem je, aby správce rozuměl nejen tomu, *jak* příkaz spustit, ale hlavně *co* jeho provedení znamená pro systém jako celek. Tam, kde je to potřeba, jsou uvedena bezpečnostní upozornění a kontrolní kroky (potvrzení, kontrolní otázky), které mají snížit riziko chyby obsluhy.

Na co je nástroj určen:

- **Monitoring stavu** (`--get_active`) – rychlá kontrola, zda je systém aktivní a vykonává přidělování času.
- **Přehled časů uživatelů** (`--list_user_times`) – zobrazení tabulky se jmény, počátečním časem, uplynulou dobou a zbývajícím časem; včetně formátovaného výstupu a zvýraznění kritických stavů.
- **Evidence uživatelů a kategorií** (`--list_users, -list_categories`) – auditovatelný náhled do registru subjektů a jejich zařazení.
- **Auditní záznamy** (`--get_logs`) – zpětná dohledatelnost zásahů a rozhodnutí.
- **Aplikace časových posunů** (`--apply_user_offset, -apply_user_cat`) – přesné, dávkové zásahy do zůstatků času jednotlivců nebo skupin.
- **Rozdělení alokace** (`--split_allocated_time`) – rovnoměrná distribuce dostupné alokace mezi všechny evidované subjekty.

- **Řízení provozních režimů** (`--set_core_mode`, `-authorize`, `--set_user`) – přístup do zvýšeně nebezpečných zón řízení, včetně vícestupňového potvrzení a autentizace.
- **Aktivace/deaktivace systému** (`--set_active`) – řízené uvedení systému do aktivního stavu nebo jeho bezpečné zastavení.

Proč je opatrnost zásadní: V prostředí In-Time systém neposkytuje pouze informaci; *čas je hodnotou, která přímo určuje délku života*. Každé navýšení, snížení či reset zůstatku času proto představuje rozhodnutí s etickými i právními důsledky. Chybný příkaz, špatně zvolený parametr nebo neověřená dávková operace může znamenat reálné ztráty na životech. Z tohoto důvodu utilita záměrně vyžaduje:

- víceúrovňové potvrzení u kritických operací,
- kontrolní otázky ověřující pozornost obsluhy,
- striktní oddělení režimů (`CORE_MODE`, `AUTHORIZED_MODE`) a identifikaci operátora,
- auditní záznamy všech zásahů pro následnou kontrolu a forenzní analýzu.

Rozsah a filosofie práce s nástrojem: Manuál je psán pro zkušené správce, kteří rozumějí dopadům svých příkazů a přijímají odpovědnost za důsledky. Doporučené workflow vždy začíná diagnostikou (získání stavu, přehledu uživatelů a logů), pokračuje přípravou změn (simulace, výpočet offsetů, ověření cílové množiny) a končí *reverzibilními* kroky všude tam, kde je to možné. **Než provedete zásah**, ověřte:

1. že cíl a dopad změny jsou přesně vymezené,
2. že parametry odpovídají záměru (uživatelé, kategorie, velikosti posunů),
3. že existuje konzistentní auditní stopa a schválení časovým výborem,
4. že běží správný režim a je nastavena identita obsluhy.

Etické a provozní zásady: Nástroj má být užíván výlučně ve prospěch stability a bezpečí federace. Jakékoliv testování provádějte odděleně od produkčního systému. V rámci produkce nikdy nespouštějte hromadné operace bez předchozího omezení rozsahu (např. na malý vzorek) a bez dvojího ověření parametrů. V případě pochybností je správným postupem *neprovádět* změnu a eskalovat ji časovému výboru.

Následující kapitoly rozvádějí jednotlivé přepínače, návrhy pracovních postupů a doporučení pro bezpečný provoz. **Pamatujte:** v systému In-Time jsou čísla více než čísla; představují čas, a tím také život. Jednejte s maximální opatrností.

Kapitola 2

Uvedení do provozu

Předmluva k provozu

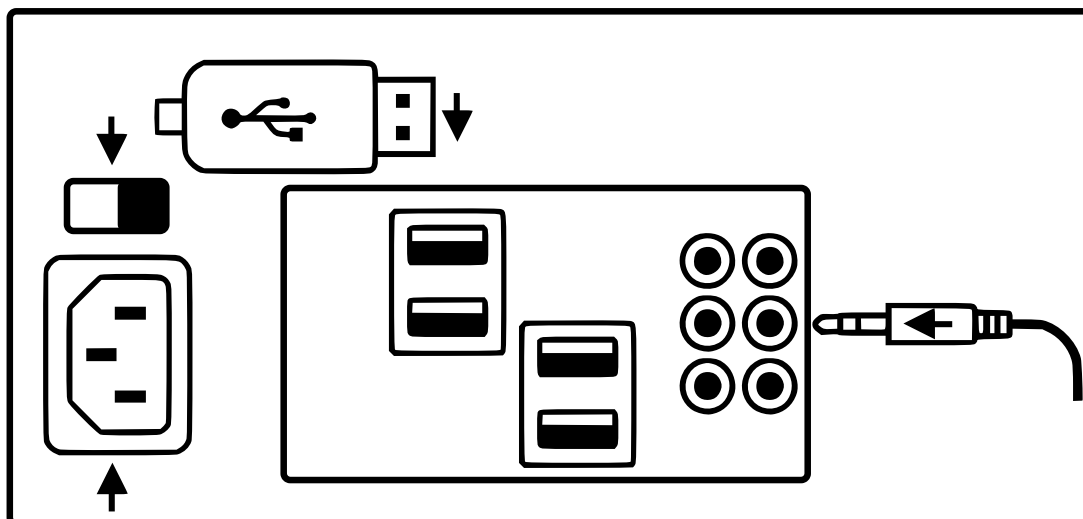
In-Time Server je navržen jako spolehlivé, průmyslově robustní řešení pro nepřetržitý provoz. Kombinuje deterministické chování serverového Linuxu s pečlivě řízenými službami, které zajišťují konzistentní distribuci času napříč celým systémem. Díky jednoduché, ale auditovatelné správě a důslednému logování poskytuje správcům *časového výboru* jistotu, že každý cyklus alokace i každá změna konfigurace je dohledatelná a reprodukovatelná. Architektura je od základu postavená na principech minimální závislosti, čitelného startu bez grafického splash screenu a preferenci textových rozhraní — aby bylo *vždy* jasné, co se v systému děje.

2.1 Zapojení

Před prvním spuštěním proveďte fyzické zapojení podle následujících kroků. Všechna připojení provádějte při vypnutém napájení serveru.

1. **Napájení:** Připojte síťový kabel **230 V AC** do zdroje v zadní části skříně serveru. Ujistěte se, že zásuvka odpovídá normě a je chráněna jističem a proudovým chráničem.
2. **Síť/bezdrát:** Zasuňte **bezdrátový adaptér** (Wi-Fi) do **USB** portu v zadní části skříně. Preferujte porty přímo na základní desce pro vyšší stabilitu.
3. **Audio výstup:** Vedení z reproduktorů (**3,5 mm jack**) zapojte do **prostřední zelené zdířky** ve **spodnější řadě** audio konektorů na zadní straně počítače (standardní *Line Out*).
4. **Volitelně:** Připojte ethernetový kabel do RJ-45 (doporučeno pro management), monitor přes HDMI/DisplayPort a klávesnici do USB.

Jakmile je vše zapojeno, systém je **připraven ke spuštění**. Doporučujeme vizuálně zkontrolovat všechny konektory a zajistit kabeláž proti náhodnému vytažení.



Obrázek 2.1: Schéma zapojení systému In-Time (doplň se).

2.2 Start

Server používá zavaděč **GRUB** a distribuci **Debian GNU/Linux**. Start je **bez grafického splash screenu**, takže na obrazovce uvidíte průběžné systémové hlášky.

2.2.1 Fáze BIOS/UEFI (po zapnutí)

Po zapnutí se objeví obrazovka *POST* základní desky (logo výrobce, test paměti, detekce disků).

- Typicky jsou dostupné klávesy **Del/F2** pro vstup do nastavení UEFI a **F11/F12** pro nabídku *Boot Menu*.
- Zkontrolujte, že bootovací zařízení je nastaveno na disk se systémem (*Linux*).
- Pokud se zde objeví varování o ventilátorech/napájení, vypněte a zkontrolujte hardware.

2.2.2 Zavaděč GRUB

Po *POST* se zobrazí textová nabídka **GRUB**.

- **Výchozí položka:** Debian GNU/Linux. Po krátkém odpočtu (obvykle 5 s) se spustí automaticky.
- **Možnosti:** Může být k dispozici **Advanced options** (recovery jádra). Není-li nutné, ponechte výchozí.
- Typická hláška: `Loading Linux ... Loading initial ramdisk ...`

2.2.3 Načítání jádra a *initramfs*

V této fázi Linux jádro inicializuje zařízení a mountuje kořenový souborový systém.

- Uvidíte strohé texty o detekci CPU, paměti, disků (NVMe/SATA), případně RAID/LVM.

- Při potížích s diskem se mohou objevit hlášky `EXT4-fs error, mount: special device not found` apod. V tom případě zkontrolujte kabeláž/UEFI nastavení.

2.2.4 systemd: spouštění služeb

Po předání řízení `systemd` začne start jednotlivých jednotek (`units`).

- Vlevo se objevují stavové značky `[OK]` / `[FAILED]` u služeb. Příklady, které běžně uvidíte:
 - `[OK] Reached target Basic System.`
 - `[OK] Started Network Manager.` nebo `Started ifupdown-pre.service`
 - `[OK] Started OpenSSH server daemon.`
 - `[OK] Reached target Multi-User System.`
 - `[OK] Started getty@tty1.service.`
- **Bez splash screenu** uvidíte průběh detailně; jedná se o standardní chování a je žádoucí pro transparentní diagnostiku.

2.2.5 Přihlašovací výzva (TTY1)

Po startu systém přejde do textové přihlašovací obrazovky:

```
Debian GNU/Linux 13 intime_server tty1
```

```
intime_server login:
```

1. Zadejte **uživatelské jméno** a potvrďte **Enter**.
2. Zadejte **heslo** (znaky se nezobrazují) a potvrďte **Enter**.

Po přihlášení se zobrazí uvítací zpráva (`motd`) a shell prompt:

```
Welcome to Debian GNU/Linux 13 (trixie)!
user@intime_server:~$
```

Poznámky a tipy:

- Pokud se **GRUB nezobrazí** a systém bootuje jiný OS, zkontrolujte pořadí boot zařízení v UEFI.
- **Dlouhé zpoždění** při `systemd` startu často souvisí s čekáním na síť. Ověřte kabeláž a konfiguraci.
- V případě `[FAILED]` u služby pokračujte analýzou přes `journalctl` a opravte konfiguraci dříve, než systém uvedete do produkce.

Důležité bezpečnostní upozornění: Tento server přímo ovlivňuje životní čas občanů. Před přechodem do aktivního režimu vždy ověřte správnost konfigurace, časových zdrojů, připojení periférií a dostupnost síťových služeb. Každá chyba v této fázi může mít následky v produkčním provozu.

Kapitola 3

Prostředí příkazové řádky

3.1 Co je *shell* a jak do něj psát

Shell (též *příkazová řádka*) je textové rozhraní, do kterého píšete příkazy. Po přihlášení do systému Debian uvidíte tzv. *prompt* (výzvu), např.:

```
user@server:~$
```

Blikající kurzor označuje místo, kam se bude vkládat text. Příkaz vytvoříte tak, že napíšete jeho název a případné argumenty/přepínače a potvrdíte klávesou **Enter**. Shell je **case-sensitive** (rozlišuje malá/velká písmena): ‘Time_Server’ není totéž jako ‘time_server’.

3.2 Základní editace a ovládání klávesnicí

Při psaní příkazů se hodí několik klávesových zkratk.

| Klávesa | Akce | Poznámka |
|------------|-----------------------|---|
| ↑ / ↓ | Historie | Prochází dříve zadané příkazy. |
| ← / → | Pohyb kurzoru | Přesun o znak vlevo/vpravo. |
| Home / End | Začátek / Konec řádku | Rychlý přesun na okraj řádku. |
| Tab | Doplňování | Automatické doplnění názvů příkazů/souborů. |

Potvrzení příkazu. Příkaz se **nevykoná**, dokud nestisknete **Enter**. Pokud příkaz čeká na vstup (např. otázka „*Proceed? [y/N]*“), odpovězte a opět potvrďte **Enter**.

3.3 Co je příkaz, argument a přepínač

Příkaz je program nebo vestavěná funkce shellu. Za názvem příkazu mohou následovat **argumenty** (hodnoty, cesty, čísla) a **přepínače** (volby), které mění chování příkazu.

```
time_server --list_user_times  
^command    ^switch (dlouhý přepínač)
```

3.4 Tipy a triky

Pokud nevíte, jaké argumenty u daného příkazu použít, většinou je dostupný přepínač **--help**, který zobrazí úhlednou nápovědu, jak program používat.

Kapitola 4

Použití příkazu `time_server`

Utility se spouští příkazem:

```
$ time_server [PŘEPÍNAČE]
```

Pouze jeden přepínač může být použit současně.

4.1 Obecné přepínače

| Přepínač | Argumenty | Popis |
|-----------------------------------|-----------|---|
| <code>--get_active</code> | žádné | Zobrazí, zda je systém aktivní (True/False). |
| <code>--list_user_times</code> | žádné | Vypíše tabulku uživatelů a jejich zbývajících času. |
| <code>--list_users</code> | žádné | Podrobný seznam uživatelů v systému. |
| <code>--list_categories</code> | žádné | Vypíše všechny výplatní třídy |
| <code>--get_logs</code> | žádné | Vypíše tabulku systémových logů. |
| <code>--get_allocated_time</code> | žádné | Vypíše celkový přidělený čas v systému. |

4.2 Modifikační přepínače

| Přepínač | Argumenty | Popis |
|----------------------------------|----------------|---|
| <code>--apply_user_offset</code> | USER_ID OFFSET | Aplikuje offset na konkrétního uživatele. |
| <code>--apply_user_cat</code> | CAT_ID OFFSET | Aplikuje offset na všechny uživatele v kategorii. |

4.3 Kritické operace

| Přepínač | Argumenty | Popis |
|------------------------|------------|--|
| --set_active | true/false | Nastaví globální stav systému (True/False). Vyžaduje potvrzení. |
| --split_allocated_time | žádné | Rovnoměrně rozdělí přidělený čas mezi uživatele. Vyžaduje potvrzení. |



Varování

Použití **kritických operací** v nástroji In-Time může vést k *narušení systému* a následnému *ohrožení lidské společnosti* se **závažnou újmou na životech**. Provádějte pouze po dvojím ověření parametrů, v režimech CORE_MODE a AUTHORIZED_MODE, se souhlasem časového výboru a s kompletní auditní stopou.

4.4 Režimy a autentizace

| Přepínač | Argumenty | Popis |
|-----------------|-----------|---|
| --set_core_mode | žádné | Aktivuje CORE režim (nebezpečné operace). Vyžaduje potvrzení a kontrolní otázku. |
| --authorize | žádné | Aktivuje AUTHORIZED režim (kritické operace). Vyžaduje CORE režim, nastaveného uživatele a heslo. |
| --set_user | USER_NAME | Nastaví aktuálního operujícího uživatele. Vyžaduje potvrzení a kontrolní otázku. |

Kapitola 5

Příklad použití

5.1 Zjistit celkový čas v systému

Celkovou časovou dotaci v rámci celé části systému lze vypsát pomocí:

Vstup:

```
time_server --get_allocated_time
```

Příklad výstupu:

```
180000
```

5.2 Výpis zbývajících časů uživatelů

Aktuální časy uživatelů, stejně jako se vypisují na tabulích, lze vypsát pomocí:

Vstup:

```
time_server --list_user_times
```

Příklad výstupu:

Přijímám data...

| Jméno | Start | Offset(s) | ... | Výsledný čas(fmt) |
|------------------|---------------------|-----------|-----|-------------------|
| Paul Gingerbread | 2025-08-10T22:49:50 | 7547320 | ... | 72:05:09:25 |
| Sim Wade | 2025-08-12T12:00:37 | 1411200 | ... | 2:17:51:32 |
| Slim Slime | 2025-08-23T09:01:04 | 453200 | ... | 2:12:45:19 |

5.3 Výpis registru uživatelů

Tabulku všech uživatelů v systému vypíšeme pomocí:

Vstup:

```
time_server --list_users
```

Příklad výstupu:

Přijímám data...

| ID | Tag | Jméno | Acr | Offset(s) | ... | Aktivní |
|----|----------------------|------------------|-----|-----------|-----|---------|
| 1 | 04:2C:0F:22:0D:6B:80 | Paul Gingerbread | * | 7547320 | ... | 1 |
| 2 | BC:1E:11:35 | Sim Wade | P | 1411200 | ... | 1 |
| 3 | 00:36:8B:3A | Slim Slime | G | 453200 | ... | 1 |

5.4 Přidat/odebrat čas uživateli

Pomocí příkazu výše zjistíme z tabulky *ID* uživatele; to použijeme v tomto příkazu, který uživateli 1 (Karel) přidělí čas 1800 s:

Vstup:

```
time_server --apply_user_offset 0 1800
```

Příklad výstupu:

| ID | Tag | Jméno | Acr | Offset(s) | ... | Aktivní |
|----|----------------------|------------------|-----|-----------|-----|---------|
| 1 | 04:2C:0F:22:0D:6B:80 | Paul Gingerbread | * | 7547320 | ... | 1 |
| 2 | BC:1E:11:35 | Sim Wade | P | 1411200 | ... | 1 |
| 3 | 00:36:8B:3A | Slim Slime | G | 453200 | ... | 1 |
| 4 | 7C:97:A4:34 | Guess Feinmount | D | 1785626 | ... | 1 |

S kategoriemi je proces obdobný.

5.5 Vstup do chráněného režimu

Pro vstup do chráněného režimu se používají přepínače pro režimy. Hierarchicky na sebe navazují následovně: *CORE_MODE* umožňuje vstup do *SET_USER* a ten následně přihlášení správce pomocí:

Vstup:

```
time_server --authorize
```


5.6 Spustit systém

Po aktualizacích na novou verzi systému je často potřeba systém znovu spustit v nové verzi. K tomu lze použít následující příkaz. Z dokumentace výše je zřejmé, že lze místo `true` použít i `false`, avšak toto se z hlediska rizika trvalého zastavení systému silně nedoporučuje.

Vstup:

```
time_server --set_active true
```

Příklad výstupu:

Nastavování...

Synchronizace...

Poslední úpravy...

Operace proběhla úspěšně

Stav systému nastaven na: True

