

## परिचय

२०२०-२०२१ परियोजना का हिस्सा, एक शक्तिशाली औजार है जो २०२०-२०२१ पर एक २०२०-२०२१ पर्यावरण लाता है। यह विकसक और पावर यूज़र्स को २०२०-२०२१ पर २०२०-२०२१ कमांड और २०२०-२०२१ औजारों का उपयोग करने की अनुमति देता है, जिससे क्रॉस-प्लेटफॉर्म प्रोजेक्ट वर्कफ्लो को सरल बनाया जाता है या २०२०-२०२१ की सुखदता प्रदान की जाती है। इस ब्लॉग पोस्ट में, हम २०२०-२०२१ के बारे में गहराई से जानेंगे, इसे अन्य २०२०-२०२१ टर्मिनलों से अलग कैसे है (और २०२०-२०२१ के साथ तुलना को स्पष्ट करेंगे), आमतौर पर उपयोग किए जाने वाले कमांडों का पता लगाएंगे, और उपयोगकर्ताओं को पथ हैंडलिंग के अलावा अन्य पहलुओं पर विचार करने चाहिए।

## \_\_\_\_\_64 क्या है?

64 का मतलब है “64-वर्षीय विकास पर्यावरण”। यह एक विकास पर्यावरण है जो एक सूट 64 औजारों और लाइब्रेरियों प्रदान करता है, जिससे उपयोगकर्ता 64-वर्षीय पर 64-वर्षीय सॉफ्टवेयर बनाना और चलाना सकते हैं। यह विशेष रूप से उन विकसकों के लिए मूल्यवान है जो 64-वर्षीय के लिए सॉफ्टवेयर को कम्पाइल करना चाहते हैं जबकि 64-वर्षीय औजारों और कमांडों की परिचितता और शक्ति का आनंद लेना चाहते हैं।

अन्य इंटर्नेशनलों से अंतर

जब 64 बिटों को 64 बिट पर उपलब्ध टर्मिनलों जैसे 64 बिट या 64-बिट-नेटवि टर्मिनलों के साथ तुलना की जाती है, तो कई अंतर सामने आते हैं। (ध्यान दें: 64 एक 64 बिट के लिए टर्मिनल एमुलेटर है, नहीं 64 बिट, इसलिए हम अपने तुलना को 64-बिट-नेटवि टर्मिनलों पर केंद्रित करेंगे।)

## 1. शैल पर्यावरण

- ८) **प्र० ६४:** इनका उपयोग करता है, जो अधिकांश ००००-०००० सिस्टमों पर डिफॉल्ट है, जिससे ०००० स्क्रिप्ट और कमांडों का नेटिव उपयोग संभव होता है।
  - ९) **प्र० ६५:** ०००.००० पर निर्भर करता है, अपने अलग-अलग कमांड सेट और स्क्रिप्टिंग भाषा के साथ।
  - १०) **प्र० ६६:** एक अधिक उन्नत, ०००००००-केनदरति शेल प्रदान करता है, अपने सिंटैक्स और क्षमताओं के साथ।

## 2. कमांड सेट

- **लाइन 64:** यहाँ कमांडों जैसे ls, grep, sed, और awk का समर्थन करता है, जो लाइनेस विद्युत या लाइनेसियल में डिफ़ॉल्ट रूप से उपलब्ध नहीं हैं।
  - **लाइनेस लाइनेस:** लाइनेस-खास कमांडों तक सीमित है (जैसे, dir, copy).
  - **लाइनेसेसेस:** कुछ लाइन कमांडों को नकल करने के लिए एलीस और मॉड्यूल शामिल हैं, लेकिन लाइन 64 एक अधिक वास्तविक और व्यापक लाइन अनुभव प्रदान करता है।

### 3. फ़ाइल सिस्टम

- **प्रैक्टिस 64:** एक ड्राइवों को / के नीचे मैप करता है (जैसे, C:\ / c/ बन जाता है), जिससे एक-एक पथ नाविगेशन संभव होता है।

■ **प्रैक्टिस 65:** एक फ़ाइल को एक विशेष स्थान पर उपयोग करते हैं। बैक्स्लैश के साथ (जैसे, C:\path\to\file).

## 4. विकास औजार

- **मोडॉल64:** यह जैसे कम्पाइलर बंडल करता है, जो एक मोडॉल-मोडॉल पर्यावरण में स्रोत से सॉफ्टवेयर बनाने के लिए आवश्यक हैं।
- **मोडॉलेट्रो मोडॉलरो मोडॉलेट्रोमोडॉलरो:** इन औजारों को डिफॉल्ट रूप से नहीं मिलते, हालांकि उन्हें अलग से जोड़ा जा सकता है; मोडॉल64 एक अधिक एकीकृत समाधान प्रदान करता है।

सारांश में, मोडॉल64 मोडॉलरो पर एक मोडॉल-मोडॉल अनुभव प्रदान करता है, जो मोडॉलेट्रो मोडॉलरो और मोडॉलेट्रोमोडॉलरो के मोडॉलरो मोडॉल-मोडॉल केन्द्रित फोकस से तेजी से अलग होता है।

## मोडॉल64 में आमतौर पर उपयोग किए जाने वाले कमांड

मोडॉल64 का मोडॉल-मोडॉल पर्यावरण एक समृद्ध कमांड सेट का समर्थन करता है। यहाँ कुछ आवश्यक हैं:

### 1. डायरेक्टरी में नाविगेशन

- `pwd`: वर्तमान कार्यकारी डायरेक्टरी को प्रदर्शित करता है (जैसे, `/c/users/yourname`).
- `cd <directory>`: निर्दिष्ट डायरेक्टरी में बदल जाता है (जैसे, `cd /c/projects`).
- `ls`: डायरेक्टरी सामग्री को सूचीबद्ध करता है (ध्यान दें: `ls मोडॉल64` में `dir` के साथ एलीस है, मोडॉल व्यवहार को नकल करता है).
- `ls -l`: विस्तृत सूची प्रदान करता है (भी `dir` के साथ विकल्पों के साथ एलीस है).

### 2. फ़ाइल और डायरेक्टरी प्रबंधन

- `mkdir <directory>`: एक नया डायरेक्टरी बनाता है (जैसे, `mkdir myfolder`).
- `rm <file>`: एक फ़ाइल को हटाता है (जैसे, `rm oldfile.txt`).
- `rm -r <directory>`: एक डायरेक्टरी और उसके सामग्री को पुनरावृत्ति से हटाता है.
- `cp <source> <destination>`: फ़ाइलें या डायरेक्टरी को कॉपी करता है.
- `mv <source> <destination>`: फ़ाइलें या डायरेक्टरी को हटाता या पुनर्नाम करता है।

### 3. फ़ाइल देखना और संपादित करना

- `cat <file>`: एक फ़ाइल के सामग्री को दिखाता है (जैसे, `cat notes.txt`).
- `less <file>`: एक फ़ाइल को पृष्ठ द्वारा देखता है.
- `nano <file>`: एक फ़ाइल को संपादित करने के लिए मोडॉल टेक्स्ट एडिटर में खोलता है।

### 4. खोज और फ़िल्टर

- `grep <pattern> <file>`: एक फ़ाइल में एक पैटर्न खोजता है (जैसे, `grep "error" log.txt`).
- `find <directory> -name <pattern>`: एक पैटर्न से मिलने वाले फ़ाइलों को खोजता है (जैसे, `find /c -name "*.txt"`).

## 5. विकास कमांड

- gcc <source.c> -o <output>: एक C प्रोग्राम को कम्पाइल करता है (जैसे, gcc main.c -o main.exe).
- make: एक Makefile का उपयोग करके सॉफ्टवेयर बनाता है.
- git <command>: Git वर्सन कंट्रोल कमांडों को अंजाम देता है (जैसे, git clone <repo>).

ये कमांड केवल बर्फ के टिप हैं—Windows 64 एक व्यापक रूप से औजारों के एकीकरण का समर्थन करता है, जिससे यह अत्यधिक बहुमुखी बन जाता है।

## अन्य पहलुओं पर विचार करना चाहिए

पथ हैंडलिंग के अलावा, Windows 64 का उपयोग करते समय कई कारकों पर ध्यान देना चाहिए:

### 1. पर्यावरण चर

- Windows 64 चर जैसे PATH, HOME, और SHELL को बनाए रखता है। उन्हें echo \$PATH के साथ देखें या export PATH=\$PATH:/new/path के साथ उन्हें संशोधित करें।
- सावधानी बरतें, क्योंकि परिवर्तन कमांड और प्रोग्राम व्यवहार को प्रभावित कर सकते हैं।

### 2. पैकेज प्रबंधन

- Windows 2, जो Windows 64 शामिल करता है, pacman पैकेज प्रबंधक का उपयोग करता है। औजारों को pacman -S <package> के साथ इंस्टॉल करें (जैसे, pacman -S gcc).
- नियमित अपडेट (pacman -Syu) आपके पर्यावरण को अद्यतन रखते हैं।

### 3. फ़ाइल अनुमतियाँ

- Windows 64 रूप से अनुमतियों (जैसे, chmod के माध्यम से) का नकल करता है, लेकिन Windows 64 में उन्हें नेटिव रूप से समर्थन नहीं मिलता, जिससे कार्यान्वयन के साथ अप्रत्याशित व्यवहार हो सकता है।
- अनुमतियों में परिवर्तन के प्रभाव Windows 64 पर सीमित हो सकते हैं।

### 4. प्रदर्शन

- एमुलेशन लेयर कुछ ऑपरेशंस को नेटिव Windows 64 औजारों से धीरे बना सकता है।
- प्रदर्शन-प्रधान कार्यों के लिए, नेटिव विकल्पों या वर्कफ्लो ऑप्टिमाइजेशन पर विचार करें।

### 5. Windows 64 के साथ एकीकरण

- Windows 64 कार्यक्रमों को सीधे चलाएं (जैसे, notepad.exe Windows 64 खोलता है)।
- Windows 64 प्रोग्रामों को अर्जुमेंट्स पास करते समय पथ परिवर्तन समस्याओं का ध्यान रखें।

## निष्कर्ष

Windows 64 और Windows 64 पर्यावरणों को जोड़ने में एक गेम-चेंजर है। इसका Windows 64 और Windows 64 कमांड सेट विकसकों और पावर यूज़र्स को Windows 64 पर बिना किसी परेशानी के काम करने की अनुमति देता है, विशेष रूप से क्रॉस-प्लेटफॉर्म विकास या कमांड-लाइन टास्क्स के लिए। जबकि पथ हैंडलिंग, अनुमतियाँ, और प्रदर्शन जैसे नुकीले पहलुओं पर ध्यान देना आवश्यक है, Windows 64 एक मजबूत और लचीला प्लेटफॉर्म बना रहता है।

इसकी पूरी क्षमता को अधिकतम करने के लिए, इसके कमांड और विशेषताओं का हाथों से पता लगाएं। औजारों के साथ प्रयोग करें, दस्तावेज़ देखें, और आवश्यकता के अनुसार ऑनलाइन संसाधनों का उपयोग करें। अभ्यास के साथ, Windows 64 आपके Windows 64 पर उत्पादकता को काफी बढ़ा सकता है। खुशी से कोडिंग करें!