LinuC レベル 1 Version10.0
技術解説無料セミナー
2021/2/21 開催

主題	1.06 シェルおよびスクリプト
副題	1.06.1 シェル環境のカスタマイズ

本日の講師

インターノウス株式会社
（LPI-Japanアカデミック認定校）

竹本 季史

会社紹介、自己紹介

#LinuC学習中

■会社紹介：インターノウス株式会社

- ●人材紹介サービス、人材派遣/SESサービス、IT未経験者の教育及び就職支援サービス、法人研修サービス
- ●未経験からインフラエンジニアやプログラマーになりたい方へ、無料で研修と就職支援サービスを行っています。
  https://engineercollege.jp/lp/

■自己紹介：竹本 季史(たけもと ときふみ)

- ●IT業界で約10年間勤務後、インターノウス株式会社エンジニアカレッジ講師。
- ●これまで約700人を未経験者からエンジニアに養成。Linuxサーバー(メール、OpenSSH、シェルスクリプト、DB、監視、演習)を担当。
- ●LinuCレベル1バージョン10.0の差分教材で「仮想マシン・コンテナの概念と利用」を執筆。

LinuCレベル１／レベル２ Version10.0とは

#LinuC学習中

■LinuCとは

クラウド時代の即戦力エンジニアであることを証明するLinux技術者認定資格

✓現場で「今」求められている新しい技術要素に対応

- オンプレミス／仮想化を問わず様々な環境下でのサーバー構築
- 他社とのコラボレーションの前提となるオープンソースへの理解
- システムの多様化に対応できるアーキテクチャへの知見

✓全面的に見直した、今、身につけておくべき技術範囲を網羅

今となっては使わない技術やコマンドの削除、アップデート、新領域の取り込み

✓Linuxの範疇だけにとどまらない領域までカバー

セキュリティや監視など、ITエンジニアであれば必須の領域もカバー

#LinuC学習中

■Version10.0と従来の出題範囲の比較

	テーマ	Version 10.0	従来
LinuC-1	仮想技術	・仮想マシン／コンテナの概念 ・クラウドセキュリティの基礎	← （Version10.0で新設）
	オープンソースの文化	・オープンソースの定義や特徴 ・コミュニティやエコシステムへの貢献	← （Version10.0で新設）
	その他	→ （Version10.0で削除）	アクセシビリティ、ディスククォータ、プリンタの管理、SQLデータ管理、他
LinuC-2	仮想化技術	・仮想マシンの実行と管理（KVM） ・コンテナの仕組みとDockerの導入	← （Version10.0で新設）
	システムアーキテクチャ	・クラウドサービス上のシステム構成 ・高可用システム、スケーラビリティ、他	← （Version10.0で新設）
	その他	・統合監視ツール（zabbix） ・自動化ツール（Ansible）	← （Version10.0で出題範囲に含む）
		→ （Version10.0で削除）	RAID、記憶装置へのアクセス方、FTPサーバーの保護、他

#LinuC学習中

■本セミナーについて

- ●本セミナーは試験範囲で問われる内容の理解を深めるために
  ポイントを解説いたします。
- ●このセミナーの内容はCentOS7を前提とした内容となっています。

■受講者の想定スキルレベル

- ●LinuCレベル1の取得を目指している方

■本セミナーのゴール

- ●シェル環境のカスタマイズができるようになる。
- ●プロファイルの変更ができるようになる。

本日解説する試験範囲(1.03.1の一部)

#LinuC学習中

主題1.03：GNUとUnixのコマンド

1.03.1 コマンドラインの操作

重要度	4
概要	コマンドラインを使用して、シェルおよびコマンドと対話できる。このトピックは、bashシェルを使用することを想定している。

詳細	• 単一シェルコマンドおよび1行のコマンドシーケンスを使用する、コマンドラインでの基本的な作業の実行。
	• 定義することを含めたシェル変数の使用と変更、環境変数の参照とエクスポート。
	  ◦ set, unset, export, env, echo, 引用符
	• コマンド履歴の使用と編集。
	  ◦ history, .bash_history
	• 定義済みパス内に存在するコマンドおよび存在しないコマンドの呼び出し。
	  ◦ bash, pwd, ., 相対パス、絶対パス
	• マニュアルの参照。
	  ◦ man

本日の内容の前提となる部分があるため、一部解説します。

本日解説する試験範囲(1.06.1)

#LinuC学習中

主題1.06：シェルおよびスクリプト

1.06.1 シェル環境のカスタマイズ

重要度	4
概要	ユーザの要求に応じてシェル環境をカスタマイズできる。全体のプロファイルおよびユーザのプロファイルを変更する。

詳細	• ログイン時または新しいシェルを生成したときに、環境変数（PATHなど）を設定する。
	  ◦ /etc/bash.bashrc, /etc/profile
	  ◦ ~/.bash_profile, ~/.bash_login, ~/.profile, ~/.bashrc, ~/.bash_logout
	  ◦ ., source, lists(;, &&, ||)
	• コマンド置換を使用する。
	  ◦ alias
	• コマンドサーチパスを適切なディレクトリに設定する。
	  ◦ PATH

目次①

#LinuC学習中

- 変数とは
  - 実習①　変数の定義と確認
- シェル変数と環境変数
  - 実習②　シェル変数と環境変数の違い
- 環境変数の用途
- 環境変数PATHにパスを追加する
  - 実習③　環境変数の役割を知る
- シェル変数と環境変数のコマンドまとめ

目次②

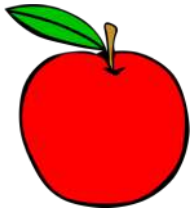
#LinuC学習中

- aliasコマンド　コマンドの別名を設定する
  - 実習④　aliasコマンドで別名の設定、確認、削除
- ログインシェルと対話型シェル
- シェル環境のカスタマイズ
- RedHat系のログイン、シェル起動、ログアウトの流れ
- sourceコマンド　現在実行中のシェルに変数やaliasを反映させる
  - 実習⑤　シェル環境のカスタマイズ

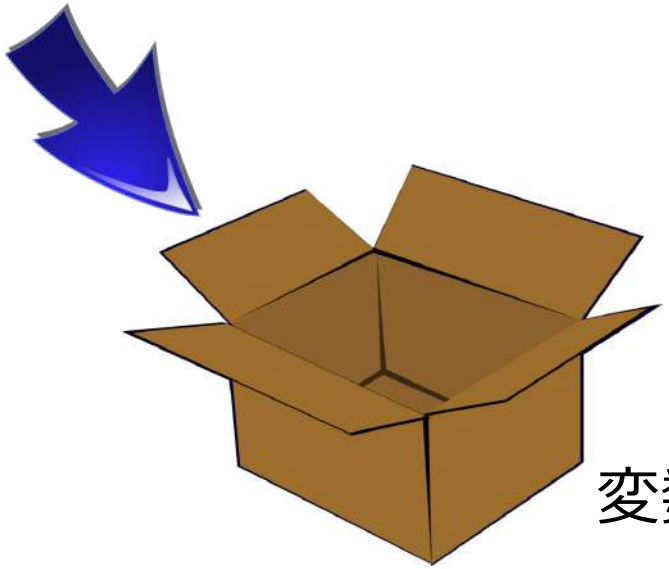
変数とは

#LinuC学習中

- 変数とは、値を保存する箱のようなものです。
- 変数にはプログラムやシェルが利用する値が格納されています。
- 変数の定義：　変数=値
- 変数の確認：　echo $変数



apple



変数box

実習①　変数の定義と確認

#LinuC学習中

1. box=apple

   左辺boxが変数です。右辺に定義する値のappleを入れます。

2. echo $box

   変数を確認するときは、echoコマンドを使用します。
   変数として参照するために、変数に$を付ける必要があります。

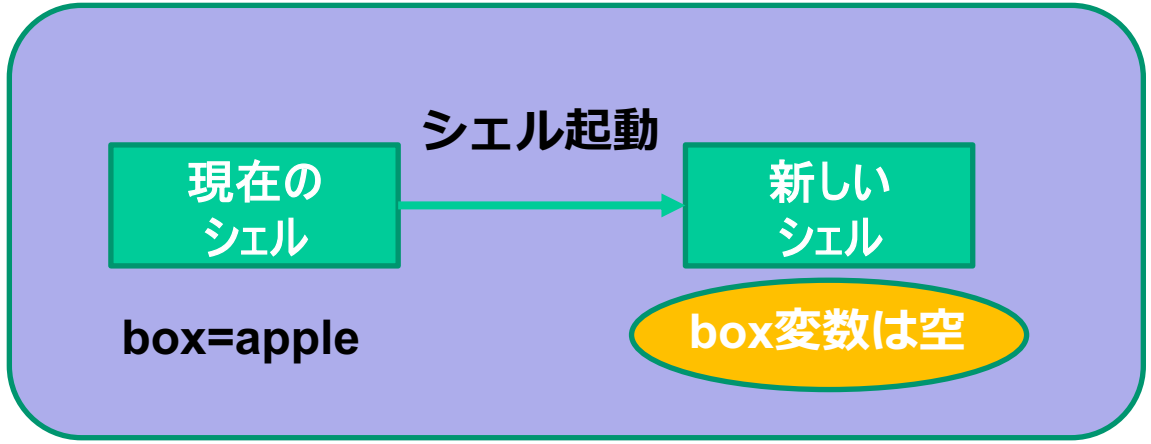
3. echo “boxの中身は$boxです。”

   このように文章内の文字列としても使うことができます。

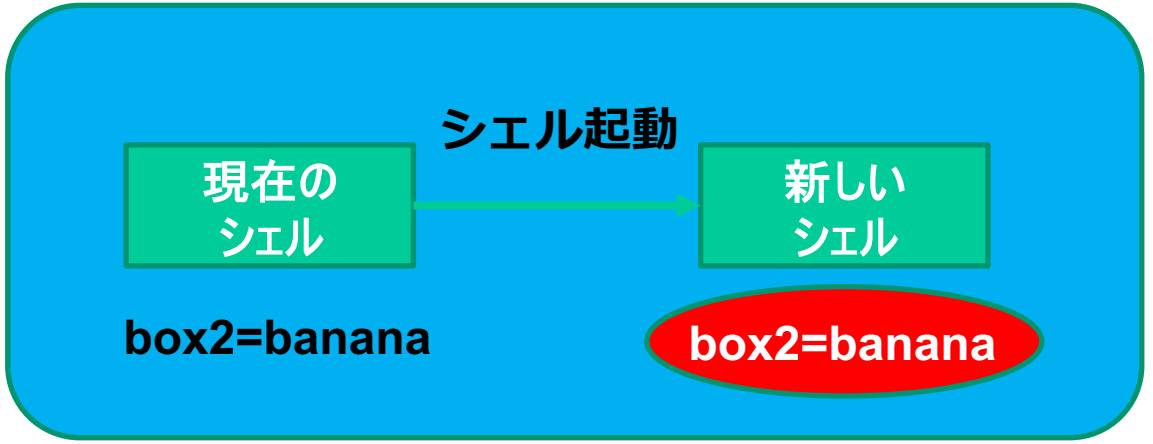
シェル変数と環境変数

#LinuC学習中

- 変数には大きくシェル変数と環境変数があります。
  - シェル変数：現在のシェルでのみ有効。
  - 環境変数　：現在のシェルで有効、さらに起動した別のシェルでも有効



シェル変数



環境変数

実習②　シェル変数と環境変数の違い

#LinuC学習中

1. box=apple　(シェル変数の定義)
2. export box2=banana　(環境変数の定義)
3. echo $box　(シェル変数の内容確認)
4. echo $box2　(環境変数の内容確認)
5. bash　(新規にbashシェルを起動)
6. echo $box　(シェル変数の内容確認)
7. echo $box2　(環境変数の内容確認)
8. exit　(5.で起動したシェルを終了)

環境変数の用途

#LinuC学習中

- 環境変数は別のシェルやシェルスクリプトを起動しても、変数を使いたいときに使います。
- 例えば、コマンドの検索パスを記述したPATH変数は環境変数であるため、別のシェルを起動しても使えるメリットがあります。
- 環境変数を定義したシェルを終了すると使用できません。
- 常に使用したい場合はログインシェル起動時に実行される/etc/profileや~/.bash_profileなどに記述します。

環境変数PATHにパスを追加する

#LinuC学習中

- PATH変数にパスを追加するには：(コロン)の後に追加するパスを記述します。

  PATH=$PATH:追加するパス

- 例えば、ホームディレクトリ配下のscriptディレクトリをPATH変数に追記するには下記を実行します。

  PATH=$PATH:$HOME/script

実習③　環境変数

#LinuC学習中

1. bash　(新しくbashを起動)
2. printenv PATH　(printenvで変数の内容が表示される＝環境変数)
3. mkdir script　(ホームディレクトリ配下にscriptディレクトリを作成)
4. echo “echo ‘This is a pen.’” > script/test.sh　(簡単なシェルスクリプトを作成)
5. cat script/test.sh　(シェルスクリプトが作成されたことを確認)
6. bash script/test.sh　(パスを指定してtest.shを実行)
7. bash test.sh　(パスを指定せずにtest.shを実行)
8. PATH=$PATH:$HOME/script　(PATH変数に$HOME/scriptを追加)
9. echo $PATH　(追加されたことを確認)
10. bash test.sh　(パスを指定せずにtest.shを実行)
11. bash　(新しくbashを起動)
12. echo $PATH　($HOME/scriptが追加されたままであることを確認)
13. exit　(11で起動したシェルを終了する)
14. echo $PATH　($HOME/scriptが追加されたままであることを確認)
15. exit　(1で起動したシェルを終了する)
16. echo $PATH　($HOME/scriptがないことを確認)

シェル変数と環境変数のコマンドまとめ

#LinuC学習中

●シェル変数と環境変数のコマンドを下表にまとめます。

	シェル変数	環境変数	説明
定義	変数=値	export 変数=値 export 変数	「export 変数=値」は環境変数を新規定義する 「export 変数」はシェル変数を環境変数にする
内容の確認	echo $変数	echo $変数	内容の確認はシェル変数と環境変数共に 「echo $変数」
一覧	set	set env printenv printenv 変数	setはシェル変数と環境変数両方表示 env,printenvは環境変数のみ表示 「printenv 変数」で指定した環境変数のみ表示
削除	unset 変数	unset 変数	変数の削除はシェル変数と環境変数共に 「unset 変数」

aliasコマンド　コマンドの別名を設定する

- aliasコマンドは、長いコマンドなどを別名として設定することで簡便にコマンドを実行するものです。

#LinuC学習中

- 設定したシェルでのみ有効です。常に使用したい場合は対話型シェル起動時に実行される/etc/bashrcや~/.bashrcに設定します。

	書式	例文	説明
設定	alias 別名='コマンド'	alias rm='rm –i'	例文では、rm –iの別名をrmと設定
確認	alias 別名	alias rm	
一覧	alias	alias	
一時無効	¥別名	¥rm	別名の先頭にバックスラッシュを付加することで、別名での実行を一時無効。
削除	unalias 別名[-a]	unalias rm unalias –a	-aオプションで全ての別名を削除。

実習④　aliasコマンドで別名の設定、確認、削除

#LinuC学習中

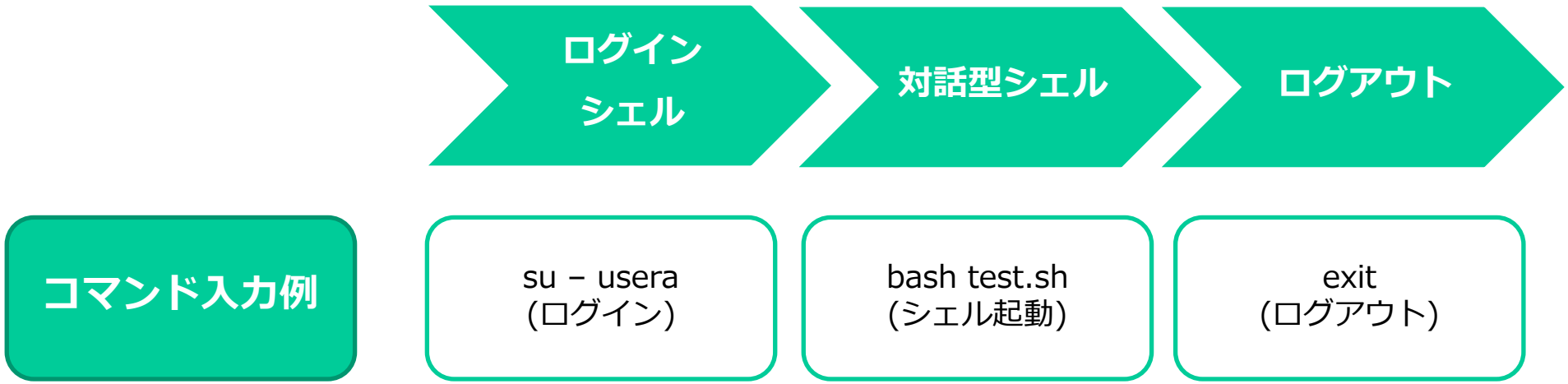
※一般ユーザーで行います。

1. alias　(別名を確認)
2. touch test1 test2 test3　(3つのテストファイルを作成)
3. rm test1　(test1が確認なしで削除される。)
4. alias rm='rm –i'　(rmの別名を設定)
5. alias rm　(rm別名の確認)
6. rm test2　(test2を削除。-iの確認表示が出る。)
7. ¥rm test3　(rm別名の先頭にバックスラッシュをつけて実行)
8. unalias rm　(rmの別名を削除)
9. alias　(別名を確認)

ログインシェルと対話型シェル

#LinuC学習中

- シェルは、大きく分けてログインシェルと対話型シェルがあります。
- ログインシェルはユーザーのログイン時に起動するシェルです。
- 対話型シェルはログイン後、シェルスクリプトの実行などで必要に応じて起動するシェルです。



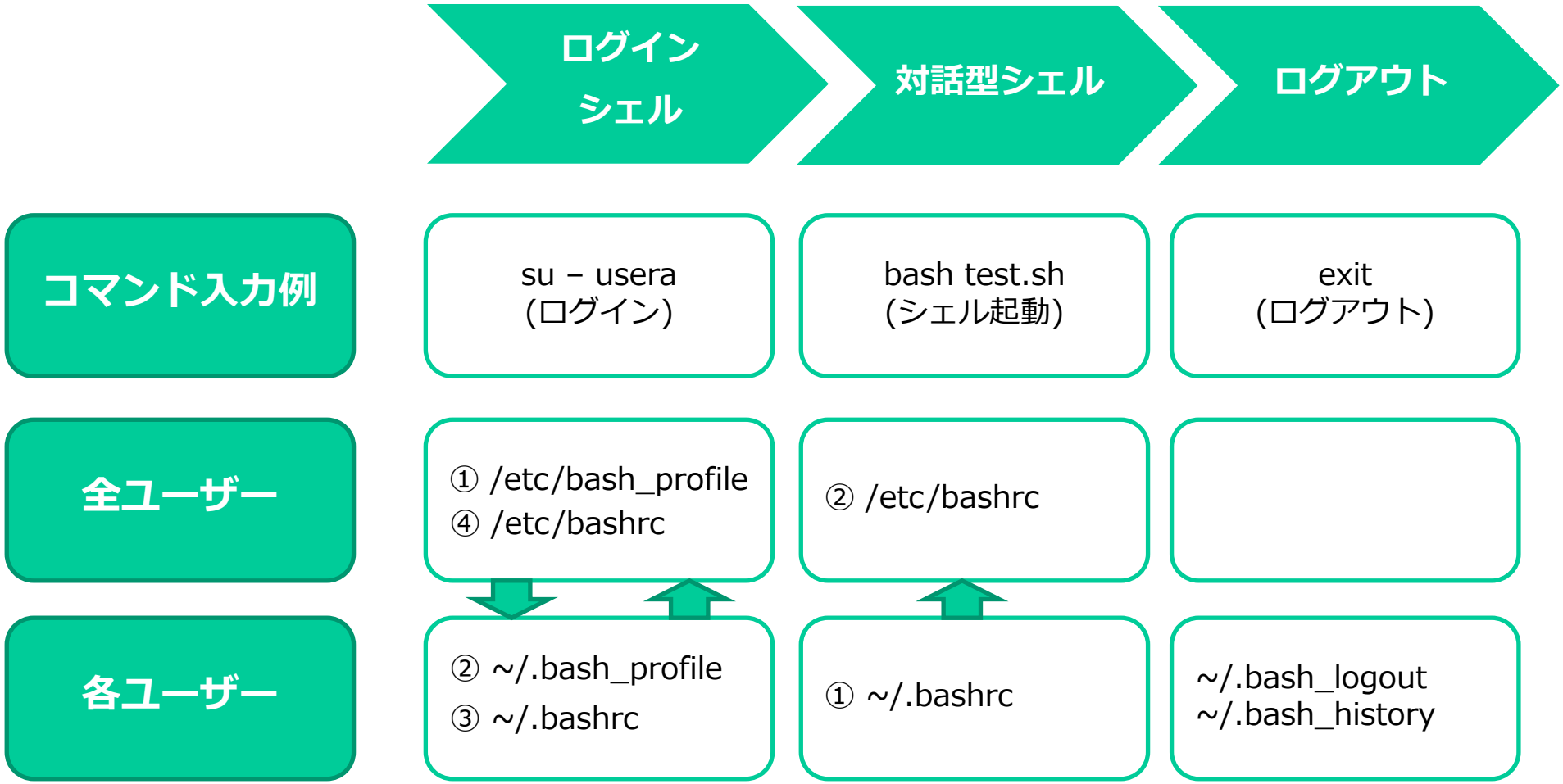
シェル環境のカスタマイズ

- シェル環境は環境変数やaliasの別名設定によりカスタマイズできますが、設定したシェルが終了してしまうと、再度設定の必要があります。
- 下記ファイルに記述することで、ログイン時や対話型シェル起動時に自動的にカスタマイズしたシェル環境を設定することができます。

適用対象ユーザー	設定ファイル	適用タイミング	ログイン時読込順	対話型シェル起動時読込順	内容
全ユーザー	/etc/profile	ログイン時	①		環境変数などを記述
	/etc/bash.bashrc	ログイン時 対話型シェル起動時	②	①	(Debian系のみ) aliasコマンドの別名設定などを記述
	/etc/bashrc	対話型シェル起動時		③	(RedHat系のみ) aliasコマンドの別名設定などを記述
各ユーザー	~/.bash_profile	ログイン時	③		環境変数などを記述
	~/.bash_login	ログイン時	③′		~/.bash_profileが存在しないとき読み込む 環境変数などを記述
	~/.profile	ログイン時	③″		~/.bash_loginが存在しないとき読み込む 環境変数などを記述
	~/.bashrc	対話型シェル起動時	④	②	aliasコマンドの別名設定などを記述
	~/.bash_logout	ログアウト時	-	-	ログアウト時に実行することを記述
	~/.bash_history	ログアウト時	-	-	コマンドの実行履歴を保存する 現在実行中のシェル終了時に書き込まれる

RedHat系のログイン、シェル起動、ログアウトの流れ

#LinuC学習中



sourceコマンド　現在実行中のシェルに変数やaliasを反映させる

#LinuC学習中

- シェル環境のカスタマイズに設定した変数やaliasはファイル変更後、次回のログイン時や次回の対話型シェル起動時から適用されます。
- 現在実行中のシェルに反映させたいときは、sourceコマンドを使用します。
- sourceコマンドは .(ピリオド)で代用できます。

コマンド	書式	例文	説明
source	source ファイル名	source .bash_profile source .bashrc	指定したファイル内の変数やaliasを現在実行中のシェルに反映
.(ピリオド)	. ファイル名	. .bash_profile . .bashrc	ピリオドとファイル名の間は半角スペース 動作はsourceコマンドと同様

実習⑤　シェル環境のカスタマイズ

#LinuC学習中

1. 一般ユーザーでログイン　(ログインシェル起動)
2. vi .bash_profile　(PATH変数に$HOME/scriptを追加)
3. echo $PATH　(変更されていない)
4. exit　(GUI環境ならログイン画面に戻る)
5. 再度一般ユーザーでログイン　(ログインシェル起動)
6. echo $PATH　(2の変更が反映されている)
7. echo "alias rm='rm -i'" >> .bashrc (alias rm='rm -i'を追記)
8. alias rm　(変更されていない)
9. bash　(対話型シェルを起動)
10. alias rm　(6の変更が反映されている)
11. exit　(9の対話型シェルを終了)
12. alias rm　(変更されていない)
13. source .bashrc
14. alias rm　(6の変更が反映されている)