

1. シェルおよびスクリプト

シェル環境のカスタマイズ/シェルスクリプト

2. ネットワークの基礎

インターネットプロトコルの基礎/基本的なネットワーク構成/基本的なネットワークの問題解決/クライアント側のDNS設定

3. システム管理

アカウント管理/ジョブスケジューリング/ローカライゼーションと国際化

4. 重要なシステムサービス

システム時刻の保守/システムのログ/メール転送エージェント(MTA)の基本

5. セキュリティ

セキュリティ管理業務の実施/ホストのセキュリティ設定/暗号化によるデータの保護/クラウドセキュリティの基礎

6. オープンソースの文化

オープンソースの概念とライセンス/オープンソースのコミュニティとエコシステム

1. シェルおよびスクリプト

シェル環境のカスタマイズ

用語: `., source, /etc/profile, ~/.bash_profile, ~/.bash_login, ~/.profile, ~/.bashrc, /etc/bashrc, ~/.bash_logout, env, export, set, unset, function, alias, PATH`

., source: 現在のシェルでファイルを実行するコマンド(bashで実行した場合は新たにシェルを立ち上げる)

/etc/profile: ユーザーログイン時に読み込まれるファイル。環境変数の設定等を行う。

~/.bash_profile: ユーザーログイン時に呼び出される。ユーザ単位の設定が記載される。PATHの設定などを主に記載。

~/.bash_login: ユーザーログイン時に呼び出される。~/.bash_profileが存在しない場合に呼び出される

~/.profile: ユーザー起動時に呼び出される。~/.bash_profileと~/.bash_loginが存在しない場合に呼び出される。または、bash以外のシェルを起動シェルにした場合に呼び出される

~/.bashrc: 非ログインシェル起動時に呼び出される (例: suにハイフン(-)をつけずにシェルを起動した場合、GNOMEなどのGUIでターミナルを起動したとき)。環境変数の設定。エイリアス、functionの設定等を記載

/etc/bashrc: 非ログインシェル起動時に呼び出される。すべてのユーザ共通の設定

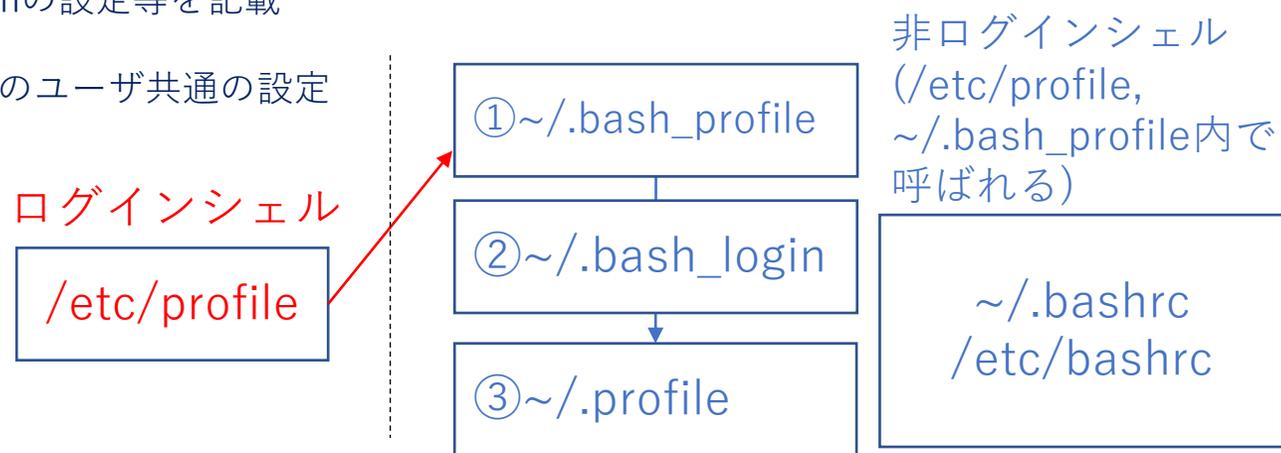
~/.bash_logout: ログアウト時に呼び出される。

PATH: コマンドを検索するディレクトリリストを作成する

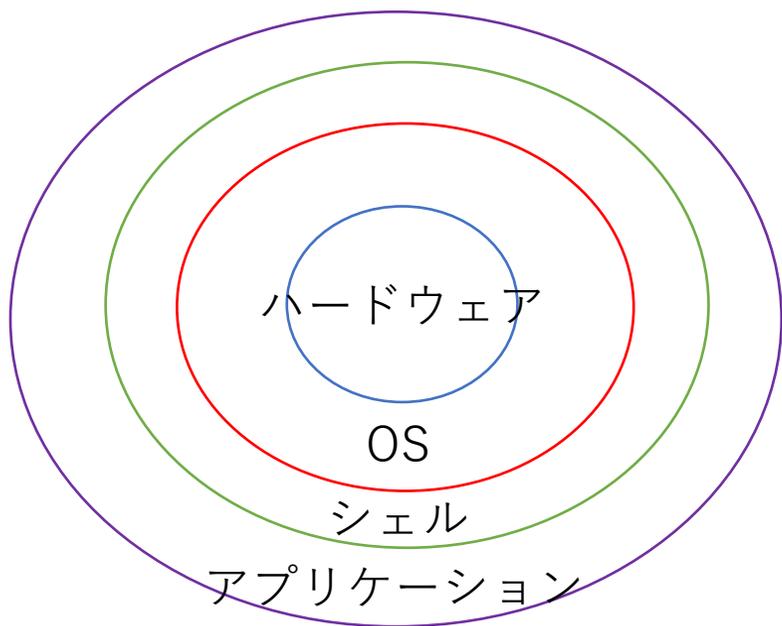
alias: コマンドを別の値で定義する

`alias ○○="××"`

function: 関数の作成



Linuxの仕組み



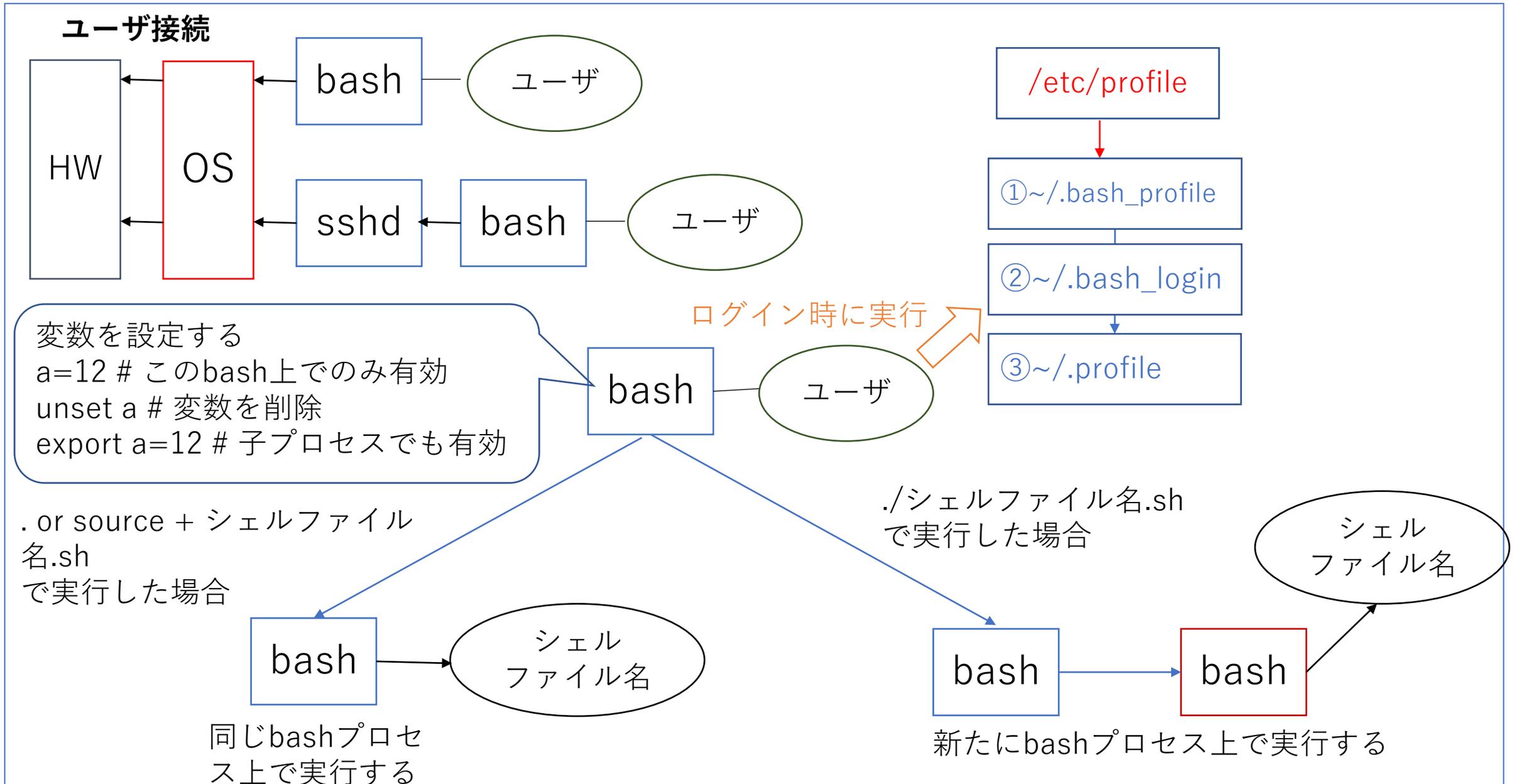
ハードウェア: CPU、メモリ、ディスクなどの物理的なデバイス

OS: ハードウェアにアクセスする

シェル: bashなど。ユーザとのインタフェース（接続部）で、コマンドの入力を受け付けて、カーネルに処理内容を伝えカーネルからの戻り値をユーザに返す

アプリケーション: シェル上に動作しているアプリケーション。ブラウザ、テキストエディタ等

Linuxの仕組み



1. シェルおよびスクリプト

シェル環境のカスタマイズ/シェルスクリプト

2. ネットワークの基礎

インターネットプロトコルの基礎/基本的なネットワーク構成/基本的なネットワークの問題解決/クライアント側のDNS設定

3. システム管理

アカウント管理/ジョブスケジューリング/ローカライゼーションと国際化

4. 重要なシステムサービス

システム時刻の保守/システムのログ/メール転送エージェント(MTA)の基本

5. セキュリティ

セキュリティ管理業務の実施/ホストのセキュリティ設定/暗号化によるデータの保護/クラウドセキュリティの基礎

6. オープンソースの文化

オープンソースの概念とライセンス/オープンソースのコミュニティとエコシステム

1. シェルおよびスクリプト

シェルスクリプト

用語: function, #!, for, while, \$?, exit, if, case, \$#, \$@, \$1...\$n, PATH, bash -x

function: 関数を定義する

```
function hello(){  
    プログラム  
}
```

#!: シェルスクリプトのファイルの冒頭に記述して、どのシェルを利用するのか指定する。

例) #!/bin/bash, #!/bin/sh, #!/bin/zsh

for, while: ループ文で利用する

for var in 1 2 3 4 5 # 変数varに1,2,3,4,5を代入していきます

while [文字列の比較] ((数値の比較)) # nが10以下の場合にwhileの中の処理を実行

do

プログラム

done

if, test: 制御文で利用する

① if test 条件文; ② if [条件文]; ③ if [[条件文]]; # ifの条件式の書き方3通り

= != #文字列で等しいか等しくないか

-eq(=), -ne(!=), -lt(<), -le(≤), -gt(>), -ge(≥) # 数値比較

1. シェルおよびスクリプト

シェルスクリプト

用語: function, #!, for, while, \$?, exit, if, case, \$#, \$@, \$1...\$n, PATH, bash -x

case: 値に応じて実行する文を変える

case \$var in # \$varはチェックする変数

パターン1) # \$varがパターン1に該当する場合に実行される

プログラム1 ;;

*) # \$varがパターン1にもパターン2にも該当しない場合に実行される

プログラム3 ;;

esac

\$?: 直前に実行したコマンドの戻り値(成功の場合0, 失敗の場合は0以外の値を返す)

\$#: 引数の数を返す

\$@: 全引数を表示

\$1, \$2, \$3...\$n: #第1引数、第2引数、第3引数、第2引数を表示

デバッグモードで実行する方法

bash -x ./hello.sh # デバッグで実行

#!/bin/bash -x # 頭に-xを付けるとデバッグを実行する

プログラム内で動的にデバッグにするか変更する方法

set -x # デバッグモード

set +x # 元に戻る