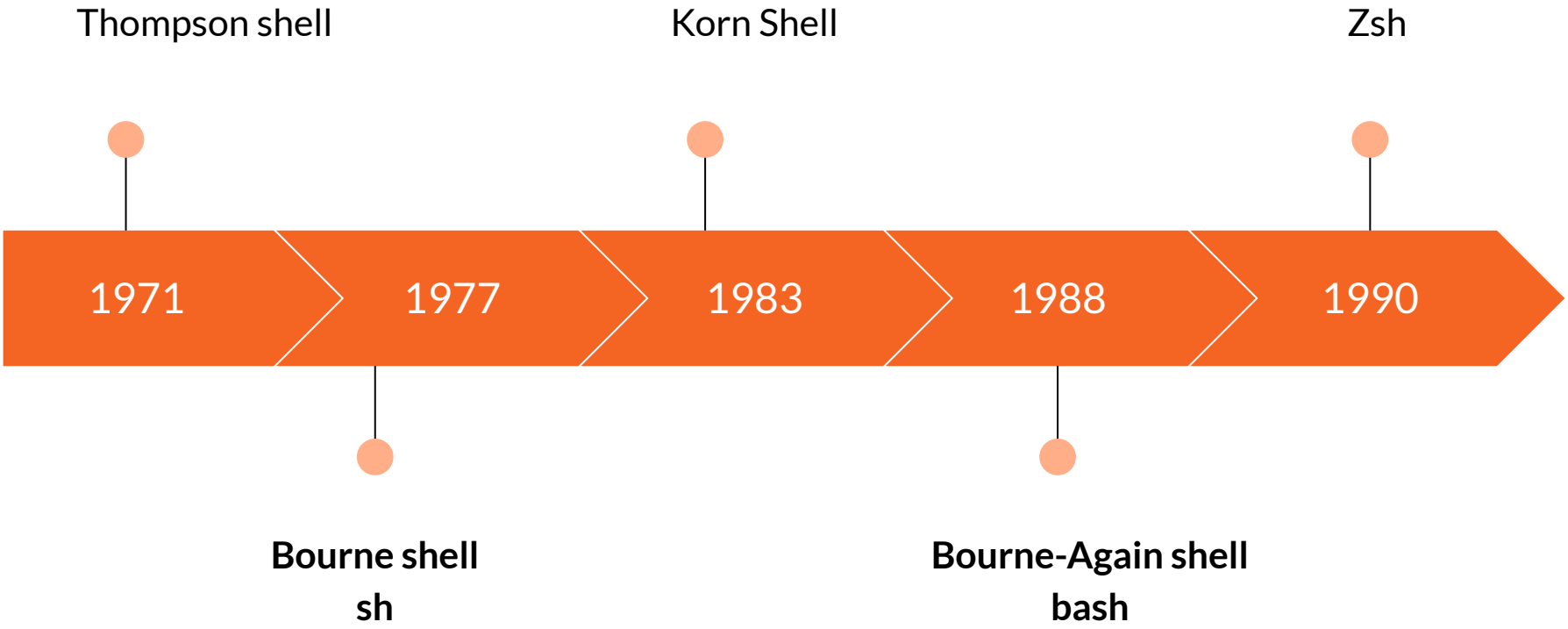
Shell Linux

Bases, Vi, Cron

Shell

- Linux
- interpréteur de commandes
- fichiers script exécutables


Thompson shell
Korn Shell
Zsh
1971
1977
1983
1988
1990
Bourne shell
sh
Bourne-Again shell
bash

Shell versus UI

- Automatisation des tâches (batch) ;
- SSH / Telnet Contrôle à distance
- Uniforme
- Stable
- Faible coût en ressources

Mais bien sûr plus “low level”

vi / vim

éditeur de texte de base

vi <nomdufichier>

Commandes:

- i -> insérer
- :w -> écrire
- :q -> quitter
- Echap : quitte du mode édition vers le mode commande

Il y a TOUJOURS vi ou vim !

Redirections

- | : redirection du flux de la sortie d'une commande vers l'entrée d'une autre
  - ls -l | wc -l
- > : remplace du contenu généré vers une entrée (la plupart du temps un fichier)
  - ls -l > ls1.out
- >> : ajoute du contenu
  - ls -l >> ls1.out

Scripting

Fichier

- nom *.sh (norme)
- exécutable
  - chmod +x <nomdufichier>
- la première ligne (shebang) définit le shell à utiliser pour l'exécution
  - #!/bin/bash

Exemple

$ vim test.sh

#!/bin/bash
echo $PATH

Echap + :wq + entrée

$ chmod +x test.sh
$ ./test.sh

Paramètres

script.sh arg1 arg2

$0 = script.sh
$1 = arg1
$2 = arg2

Variable assignation & affichage

```
message='ohé'

echo $message
```

Variable & guillemets

- simple guillemet ‘
  - aucune interprétation
- double guillemet “
  - interprète les variables
- guillemet simple inversé `
  - exécute le contenu

conditions

```
if [ test ]
then
    echo "C'est vrai"
fi
if [ test ]
then
    echo "C'est vrai"
else
    echo "C'est faux"
fi
```

if [$i -ge 20] ....
if [$# -ge 1] && [$1 = 'koala'] ...

```
case $1 in
    "test1")
        echo "test1"
        ;;
    "test")
        echo "test"
        ;;
        echo "fail"
        ;;
esac
```

boucles

```
while [ test ]
do
    echo 'Action en boucle'
done
```

```
for variable in 'val1' 'val2' 'val3'
do
    echo "$variable"
done
```

```
for i in `seq 1 10`;
do
    echo $i
done
```

Fonctions

doTest()

{
echo “$1”
echo “ok !”
return “super”
}

doTest “ok ?”
echo $?

$0 : nom de la fonction
$1 $2… : valeurs des arguments
$? : valeur du retour

Commandes “de bases”

ls : liste les fichiers

cd : change de répertoire

cp : copie

mv : déplace

rm : supprimer

mkdir : crée un répertoire

chmod : change de droit

chown : change de propriétaire

ssh : se connecter en ssh

scp : copier via ssh

rsync : synchorniser

sed : remplacement de texte

grep : extraction de texte basé sur une recherche

find : cherche un fichier

test : teste l’égalité, l’existence...

Planification

Cron

Edition avec crontab -e

- minute
- hour
- day of month
- month
- day of week

http://crontab.guru

Entrée Cron + exemple

mm hh jj MMM JJJ [utilisateur] tâche > log

* * * * 1 /root/script/like-a-######-morning.sh

21 30 24 12 * echo "OH OH OH! I have a machine gun !"

Linux sous Windows ?

Sous-système Windows pour Linux

couche de compatibilité permettant d'exécuter des fichiers pour Linux à partir d'un système Windows.

- pour développeur
- harmonisation du code
- pour une utilisation d’administration et de déploiement principalement