Systèmes informatiques

Franck Guingne,
sur la base du cours d'Olivier Lecarme

Cours Licence 1; Semestre 2

2009–2010

Neuvième cours : Personnalisation de l'environnement graphique

Plan en cours

Retour sur les principes de X

- le système de fenêtrage X sépare l'activité d'affichage de l'activité de calcul, grâce à la relation client-serveur
- dans une utilisation typique, deux ordinateurs sont connectés
- l'ordinateur local fait fonctionner un processus fondamental, le *serveur X* :
  - dévolu à l'affichage et à la communication sur un *visuel* relié à l'ordinateur local
  - un visuel rassemble un écran, un clavier et un dispositif de pointage
  - un même serveur X peut servir plusieurs visuels, avec plusieurs écrans chacun

Suite des principes de X

- l'ordinateur distant fait fonctionner un *client X*, processus qui communique avec le serveur à travers le réseau :
  - requêtes d'affichage sur un visuel donné :
    - afficher une fenêtre de géométrie et position données
    - afficher une chaîne de caractères dans une fenêtre
    - afficher un segment de couleur donnée dans une fenêtre
    - etc.
  - réception d'événements attachés à des fenêtres :
    - clic de la souris
    - frappe d'un caractère
    - démasquage de la fenêtre par une autre
    - etc.

Le protocole de X

- le protocole de X s'appuie sur TCP et IP
- utilisation du port 6000
- les *requêtes* sont des messages courts transitant sur le réseau (ou localement)
- elles sont indépendantes des deux ordinateurs en jeu
- quand un *événement* se produit, le serveur X :
  - détermine quels sont les processus clients concernés
  - envoie à chacun notification de cet événement
- ce mécanisme permet à n'importe quel ordinateur d'envoyer une requête à n'importe quel serveur X
- c'est une nécessité pour qu'on puisse exécuter un programme sur un ordinateur distant et lui faire afficher ses résultats dans une fenêtre sur l'ordinateur local

Nécessité de l'authentification

- les requêtes ne sont pas identifiées par un utilisateur
- cela me permettrait donc d'envoyer des requêtes à un serveur X qui ne m'appartient pas
- le serveur X utilise un mécanisme d'authentification :
  - une clé cryptée est générée de manière aléatoire au démarrage du serveur
  - elle est conservée dans un fichier `$HOME/.Xauthority`
  - elle doit être copiée dans tous les environnements des processus clients
  - le serveur n'accepte que les requêtes accompagnées de la clé associée au visuel

Fonctionnement de l'authentification

- si le client tourne sur le même ordinateur que le serveur :
  - le nom de visuel est celui de l'ordinateur, avec `:0.0` au bout
  - la clé est trouvée au même endroit par le client et le serveur
- sinon il faut utiliser la commande `xauth` :
  - `xauth extract - visuel` envoie sur sa sortie standard la clé associée localement au visuel donné
  - `xauth merge -` ajoute au fichier d'authentification local la clé trouvée sur son entrée standard (le nom de visuel en fait partie)
  - `xauth extract - visuel|rsh machine xauth merge -` permet donc de copier la clé du visuel local vers la machine distante
  - on peut alors lancer des clients pour ce visuel sur la machine distante

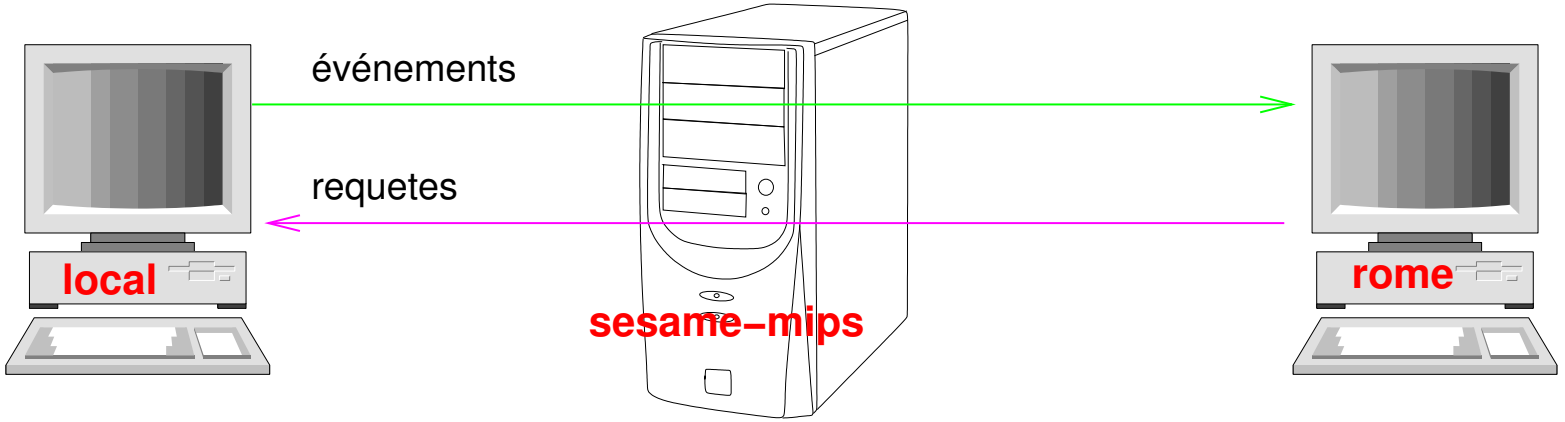
Sécurisation du protocole X

- le mécanisme de clé d'authentification n'offre qu'une faible sécurité
- les requêtes et événements restent transmis en clair sur le réseau
- l'utilisation du transfert de port du protocole SSH permet de :
  - éviter la copie à distance de la clé d'authentification
  - sécuriser toute la communication
  - on dit que les communications X passent dans un tunnel crypté
- ce mécanisme permet une connexion à une distance quelconque
- en revanche, il nécessite un débit suffisant dans les deux sens
- le mécanisme du serveur X virtuel permet de contourner cette difficulté

Fonctionnement de X sécurisé

- fonctionnement simple :
  - un pseudo-visuel est créé : `localhost:`*`numéro`*`.0`
  - le numéro part de 10, et simule l'existence d'un autre écran attaché à l'ordinateur local (*localhost*)
  - les requêtes et événements concernant ce pseudo-visuel sont transmis à travers la connexion SSH vers le bon serveur X
  - ce mécanisme peut fonctionner à travers plusieurs connexions successives
- il est normalement difficilement utilisable depuis chez vous, à cause du faible débit en sortie proposé par la plupart des fournisseurs de service

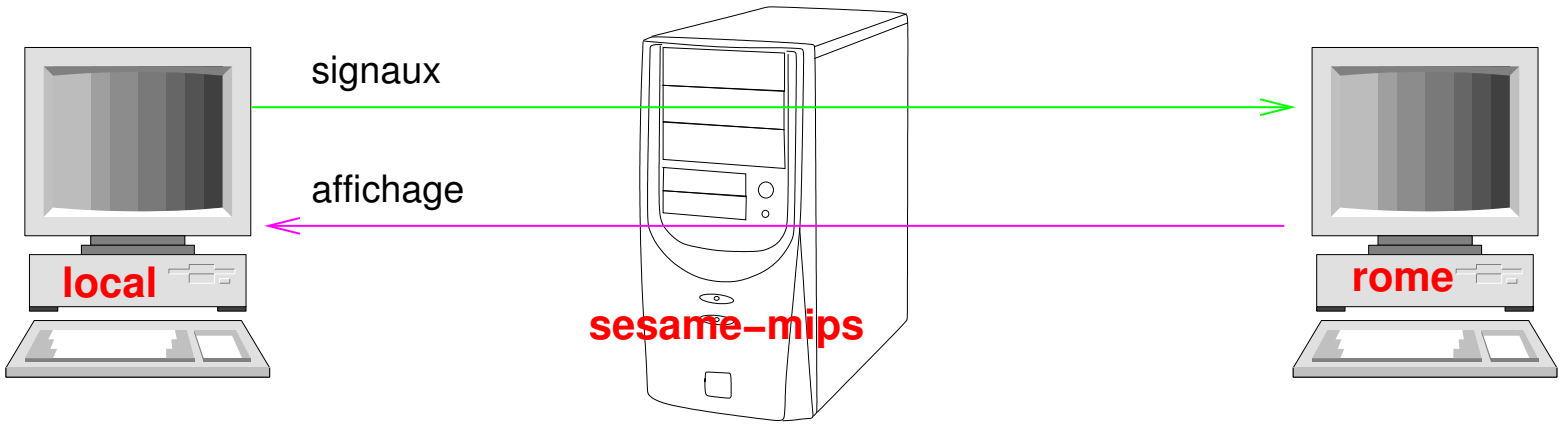
Utilisation de VNC



`ssh -X sesame-mips` — Serveur X

`ssh -X rome`

`emacs &` — Client X




```
ssh -XCL 5901:rome:5901 \
    sesame-mips
```

`ssh -X rome` — Passerelle

`vncserver` — Serveur X virtuel

Utilisation de VNC

- on établit un *tunnel SSH* entre l'ordinateur local et l'ordinateur distant, à travers la passerelle
- le port 5901 est celui de VNC pour le premier serveur
- le dialogue entre clients et serveur se fait tout entier sur l'ordinateur distant
- le tunnel SSH ne gère que l'affichage et les signaux
- dans le cas ordinaire, le tunnel SSH doit gérer la totalité du dialogue entre clients et serveur

Nom du visuel

- les clients X doivent donc connaître le nom du visuel
- en l'absence d'option l'indiquant, c'est la valeur de la variable d'environnement `DISPLAY`
- cette valeur est affectée initialement à la connexion sur un poste de travail
- dans une connexion à distance par SSH, elle est affectée au nom de pseudo-visuel si la commande `ssh` a l'option `-X` (souvent facultative)
- on peut aussi l'indiquer explicitement par l'option standard `-display` *`visuel`* :
  `rsh` *`machine`* `xterm -display` *`visuel`*
- naturellement, le client ne peut fonctionner que si l'authentification est correcte

Géométrie de la fenêtre

- la *géométrie* définit les dimensions et l'emplacement de la fenêtre
- si l'option manque :
  - le client choisit les dimensions
  - le gestionnaire de fenêtres choisit l'emplacement
- forme de l'option :
  - `[`*largeur*`x`*hauteur*`][`±*abscisse*±*ordonnée*`]`
  - la largeur et la hauteur de la fenêtre sont mesurées :
    - en pixels pour les fenêtres graphiques
    - en caractères pour les fenêtres textuelles (Xterm, Emacs)

Suite de la géométrie

- l'abscisse et l'ordonnée donnent la distance en pixels depuis le bord de l'écran :
  - `+`*`abscisse`* depuis le bord gauche pour la bordure gauche
  - `-`*`abscisse`* depuis le bord droit pour la bordure droite
  - même chose pour ±*ordonnée*
- exemple : `xclock -geometry 75x75-0+0`
  - affiche une horloge carrée de 75 pixels de côté
  - placée dans le coin en haut et à droite de l'écran

Autres options standard

- `-background` *`couleur`* donne la couleur de fond de la fenêtre (peut s'abréger en `-bg`)
- `-foreground` *`couleur`* donne la couleur de premier plan de la fenêtre (couleur d'écriture ; peut s'abréger en `-fg`)
- `-title` *`titre`* donne le nom apparaissant dans la barre de titre
- etc.
- de plus, la plupart des clients ont leurs options propres

Modèles de couleurs : RGB

- les écrans actuels représentent les couleurs par superposition
- chaque pixel de l'écran comprend en fait trois pixels des *couleurs fondamentales* :
  - R = rouge
  - G = vert
  - B = bleu
- chaque pixel coloré a une intensité de 0 à 255 (en général)
- un pixel de l'écran est donc représenté par trois octets, un par couleur fondamentale
- si les trois valeurs sont nulles, le résultat est noir
- si les trois valeurs sont maximales, le résultat est blanc

Le modèle RGB (suite)

- le modèle RGB est additif :
  - pas très intuitif
  - correspond au fonctionnement des écrans
  - correspond au fonctionnement de la rétine de l'œil
- on obtient les variations de couleurs en faisant varier les trois intensités :
  - ajouter du vert à du rouge donne du jaune
  - ajouter du bleu à du rouge donne du magenta
  - ajouter du bleu à du vert donne du cyan

Nommer les couleurs

- on représente donc une couleur par un nombre sur 24 bits (6 chiffres hexadécimaux) :
  - `00FF00` vert
  - `FF0000` rouge
  - `FFFF00` jaune
- une représentation symbolique est plus parlante :
  - `230 230 250` lavender
  - `190 190 190` grey
  - `25 25 112` midnight blue
  - `173 255 47` green yellow
  - etc.
- le fichier `/usr/lib/X11/rgb.txt` contient une telle table de correspondance

Le modèle CMYK

- le modèle CMY est soustractif
- il correspond à ce que font les imprimantes
- trois couleurs fondamentales :
  - C = cyan
  - M = magenta
  - Y = jaune
- la valeur numérique d'un pixel est le complément de celle en modèle RGB
- pour économiser les encres colorées, le modèle CMYK ajoute un composant noir (*blacK*) :
  - le minimum des trois valeurs CMY est la valeur du composant K
  - il est retranché des trois valeurs

Le modèle HSV

- le modèle HSV est le plus intuitif
- les trois composants d'une couleur sont :
  - la teinte H (*hue*), angle en degrés sur le cercle des couleurs : 0 = rouge, 180 = cyan, 120 = vert, etc.
  - la saturation S, un pourcentage : c'est l'intensité de la couleur, la réduire revient à ajouter du blanc
  - la valeur V, un pourcentage : c'est la pureté de la couleur, la réduire revient à ajouter du noir
- exemples :
  - H=175,S=228,V=182
  - H=320,S=134,V=229
- l'outil `kcolorchooser` permet de construire une couleur selon ce modèle, et d'utiliser ensuite les paramètres dans le modèle RGB
- le sélecteur de couleurs et la pipette sont disponibles dans Nautilus dans /Dossier personnel/Edition/Arrière-plans et emblèmes/Couleurs/Ajouter une couleur

Polices de caractères

- les caractères affichés sur l'écran (les *glyphes*) sont dessinés par matrices de pixels
- une *police de caractères* est un ensemble de glyphes apparentés, correspondant à un alphabet
- une police est désignée par un nom symbolique long qui en donne les caractéristiques
- exemple
  `-adobe-courier-medium-r-*-*-14-100-*-*-*-*-iso8859-1`
- une caractéristique non précisée est remplacée par le joker `*`
- l'outil `xfontsel` permet de choisir une police parmi celles qui existent

Polices de caractères (suite)

- caractéristiques principales :
  - fournisseur (*foundry*) : Adobe, Bitstream, etc.
  - famille (*family*) : Charter, Courier, Helvetica, etc.
  - graisse (*weight*) : bold, medium, light, etc.
  - inclinaison (*slant*) : italique, oblique, romain
  - corps (*pixelsize*) : 10, 12, 14, etc.
  - définition (*pointsize*) : 100, 140, etc.
  - registre (*registry*) : ISO-8859
  - codage (*encoding*) : 1
- il existe des noms symboliques abrégés pour les polices les plus courantes
- les polices utilisables par chaque serveur X ne sont pas forcément les mêmes partout
- le client ne peut pas démarrer s'il n'a pas les polices demandées

Plan en cours

Principes de la paramétrisation

- paramétrer votre environnement graphique, c'est l'adapter à vos préférences
- malheureusement, beaucoup de gens utilisent ce mécanisme pour singer des environnements existants (typiquement Windows), ce qui est très inutile
- avec Unix, tout est paramétrable, de manière plus ou moins facile
- le paramétrage des clients X utilise certains fichiers de configuration
- le paramétrage du gestionnaire de fenêtres peut se faire de deux manières :
  - modifier le contenu d'un fichier de configuration
  - utiliser un ensemble de commandes interactives fournies par le gestionnaire
- la deuxième méthode est moins générale que la première, mais plus facile

Ce qui est paramétrable

- changer la couleur du fond des fenêtres d'Emacs se fait par un fichier de paramétrage du serveur X
- changer la couleur du décor des fenêtres se fait par le fichier de paramétrage du gestionnaire de fenêtres (ou par une commande interactive de ce dernier)
- changer le comportement des fenêtres quand le pointeur passe dessus (nécessité ou non de cliquer pour rendre actif) dépend aussi du gestionnaire de fenêtres
- si l'on utilise un environnement de bureau, c'est son paramétrage qui détermine la présence de menus et boutons dans des barres de menus ou *tableaux de bord*
- si l'on utilise un gestionnaire de fichiers (partie d'un environnement de bureau), c'est son paramétrage qui détermine quel programme est appelé quand on active l'icône d'un fichier

Paramétrage par le serveur X

- un client X est en bonne partie paramétré par des *ressources X*
- il s'agit de variables pouvant prendre des valeurs :
  - couleur d'un tracé
  - épaisseur d'un trait
  - choix d'une police de caractères
  - option booléenne
  - etc.
- les ressources utilisées par un client sont déterminées par sa construction : utilisation de composants graphiques appelés *widgets*
- une spécification de ressource a la forme :
  `client[.widget...].ressource : valeur`
- les widgets indiqués représentent la structure logicielle du client, expliquée dans les pages de manuel

Spécifications des ressources

- les spécification des ressources peuvent être fournies au client de plusieurs manières :
  - trouvées dans des fichiers de configuration par défaut
  - spécifiées au moment de l'appel du client par l'option standard `-xrm`
  - chargées dans le serveur X par la commande `xrdb` *`fichier de ressources`*
- le dernier mécanisme est le plus général, puisque tous les clients y ont accès, même s'ils viennent de machines distantes

Notation utilisée dans le fichier de ressources

- la notation utilisée accepte des généralisations :
  - le signe `*` représente une suite de widgets non spécifiés : `toto*foreground` spécifie toutes les ressources de ce nom pour le client indiqué
  - on peut omettre le nom du client : `*scrollbar.foreground`
  - on peut spécifier une classe de ressources plutôt qu'une instance particulière : la classe `Foreground` comprend les instances `foreground`, `cursorColor` et `pointerColor`
  - si l'on nomme l'instance particulière du client (option standard `-name`), on peut personnaliser certaines ressources :
    `polymnie*foreground` ne concerne que les clients nommés `polymnie`, par exemple les fenêtres `xterm` sur ce serveur

Exemples de paramétrages

- police de caractères pour la barre de titre des fenêtres :
  `*titlebar*Font:-*-charter-medium-r-*-*-12-*-*-*-*-*-iso8859-1`
- police de caractère par défaut :
  `*Font:-*-courier-medium-r-*-*-12-*-*-*-*-*-iso8859-1`
- géométrie et couleur d'une fenêtre Xterm nommée (option `-name`) :

```
! fenêtre ouverte sur la station
login*geometry:80x24+660+100
login*background:#eedeff
```

- paramètres pour les fenêtres d'Emacs (classe) :

```
Emacs*background: DarkSeaGreen2
Emacs*foreground: gray9
Emacs*cursorColor: OrangeRed1
Emacs*pointerColor: blue
Emacs*borderColor: SpringGreen4
Emacs*menubar.background: cyan2
Emacs*menubar.buttonForeground: navy
Emacs.menu*.background: cyan2
Emacs.menu*.buttonForeground: navy
```

Paramétrage du clavier

- un clavier est simplement un dispositif dont chaque touche envoie un signal différent
- plus précisément, il y a un signal d'appui et un signal de relâchement, avec un code numérique caractéristique de la touche
- la correspondance entre ces signaux et l'événement attendu (envoi d'un caractère, changement de mode, etc.) est faite par le serveur X
- un outil graphique permet souvent des paramétrages de ce mécanisme :
  - choisir le type de clavier
  - déterminer la fréquence de répétition
  - changer le comportement de touches spécifiques :
    - verrouillage de majuscules
    - touches spécialisées pour Windows
    - touche de composition

Paramétrage du clavier (suite)

- le client `xkeycaps` permet de :
  - vérifier les codes envoyés par chaque touche
  - les modifier immédiatement
  - produire un fichier de configuration du clavier
- ce fichier doit être lu par l'outil `xmodmap`
- on peut ainsi :
  - faire envoyer des codes précis par certaines touches
  - changer de type de clavier
  - modifier les comportements de certaines touches
- on peut aussi construire manuellement le fichier lu par `xmodmap`, mais c'est beaucoup plus difficile
- `gnome-keyboard-properties` permet aussi de gérer le clavier mais moins finement que `xkeycaps`

Choix d'environnements de bureau

- dans le monde de GNU/Linux, trois choix principaux sont possibles :
  - l'environnement de bureau GNOME, utilisé en TP
  - l'environnement de bureau KDE
  - le gestionnaire de fenêtres `fvwm2`, ou tout autre gestionnaire perfectionné, utilisé sans environnement de bureau
- quand on utilise un environnement de bureau, le gestionnaire de fenêtres est discret et peu paramétrable (`Metacity` pour GNOME)
- dans d'autres systèmes Unix, on trouve souvent :
  - l'environnement de bureau CDE, inspirateur de KDE
  - plusieurs gestionnaires de fenêtres, en particulier `fvwm2`
  - le choix est principalement affaire de goûts personnels

Paramétrage de fvwm2

- en version simple, `fvwm2` offre une paramétrisation interactive minimale
- la version `fvwm-themes` offre au contraire une paramétrisation interactive très complète
- cette version offre également le choix entre de nombreux *thèmes*, ensembles de paramètres offrant une présentation harmonisée des différents aspects
- la paramétrisation par fichier utilise un fichier `.fvwm2rc` dans le répertoire `$HOME/.fvwm`
- c'est un fichier qu'on peut modifier manuellement pour définir de manière très fine l'ensemble des comportements

Fragment de fichier de configuration pour fvwm2

```
# # Gagner de la place en aimantant entre elles les
fenêtres et les icônes
SnapAttraction 20 SameType
# # Configuration des polices
Style "*" Font -*-lucida-medium-r-*-*-12-*-*-*-*-*-*-*
Style "*" IconFont -*-lucida-medium-r-*-*-11-*-*-*-*-*-*-*
# # Options pour les écrans virtuels
DesktopSize 3x3
EdgeResistance 750 75
OpaqueMoveSize 25
# # Style de bordure des fenêtres
BorderStyle Active Solid chocolate1
BorderStyle Inactive Solid MediumPurple4 - NoInset
HiddenHandles
# # Menu des commandes les plus fondamentales
AddToMenu Principal "Principal" Title
+ ...terminal Popup Terminaux
+ ...fenetres Popup Fenetres
+ ...serveurs Popup Serveurs
```

etc.

Paramétrage de GNOME

- la paramétrisation se fait de manière interactive
- les composants paramétrables sont en particulier :
  - le gestionnaire de fenêtres Metacity, de manière très limitée
  - le gestionnaire de fichiers Nautilus
  - le fond d'écran
  - les tableaux de bord, et en particulier leurs composants
  - les comportements généraux
  - les thèmes de décor général
- sur des composants particuliers, la paramétrisation se fait comme suit :
  - le bouton 3 de la souris dans le composant fait apparaître un menu
  - on choisit l'entrée Préférences ou l'entrée Propriétés
  - on modifie des valeurs ou des options dans la fenêtre à onglets qui apparaît

Exemples de paramétrages

- presque tous partent du menu Paramètres du bureau, sous-menu Préférences
- utilisation de commandes au clavier
  - il est très souvent utile de pouvoir changer de fenêtre ou d'écran virtuel sans quitter le clavier
  - l'entrée Raccourcis clavier propose des choix pour un certain nombre d'actions importantes
- choix du thème de décor
  - entrée Thèmes
  - un certain nombre de thèmes sont prédéfinis
  - on peut en trouver d'autres sur de nombreux sites
  - un thème choisit les couleurs, la présentation des boutons de barre de titre, leur disposition, les icônes de base, etc.

Suite des exemples

- fond de l'écran
  - entrée Arrière-plan, qu'on trouve aussi dans le menu obtenu dans le fond d'écran
  - on peut choisir une image, ce qui n'est pas forcément une bonne idée
  - on peut aussi choisir une couleur et un dégradé
- comportements par défaut
  - entrée Fenêtres
  - un des comportements possible est que la fenêtre sous le pointeur devienne active
  - on peut aussi amener automatiquement au premier plan la fenêtre active, après un délai paramétrable
- apparence des menus, boutons et tiroirs
- etc.

Tableaux de bord

- un *tableau de bord* est une barre sur un côté de l'écran, munie de boutons et de mini-fenêtres
- on peut en créer de nouveaux ou en supprimer
- on peut ajouter ou enlever des composants à chaque tableau de bord :
  - menus prédéfinis
  - lanceurs d'applications spécifiques :
    - pris dans des menus existants
    - définis manuellement
  - utilitaires
    - liste des fenêtres
    - changeur de bureau (écrans virtuels)
    - monteur de disques (pour le CD)
    - indicateur de présence de courrier
    - etc.
  - etc.