

Licence L2 SdN Durée : 1, 5 heure

Date: / / 2023 Nb page(s): 3

Professeur : Weinberg

Sans document / Sans calculatrice

**EXAMEN Architecture Système** **– Semestre 2 – Session 1**

Contexte

* Associer les noms des commandes Unix à leur fonction (2 points)

|  |  |  |  |
| --- | --- | --- | --- |
| ls | • | • | Copie un (ou des) fichier(s) |
| cd | • | • | Crée une archive |
| man | • | • | Fournit de l’aide sur une commande |
| cp | • | • | Change les droits d’un fichier |
| more | • | • | Liste le contenu d’un dossier |
| tar | • | • | Crée un nouveau dossier |
| chmod | • | • | Change le dossier courant |
| mkdir | • | • | Affiche le contenu d’un fichier |

* Quel **symbole** est utilisé pour introduire une option dans une commande Unix (Il y a deux réponses possibles, vous donnerez la plus « simple ») ? (0.5 point)
* Fournir **deux** exemples d’options avec leur signification/usage pour la commande **gcc**.

(1 point)

* Par quoi commence un fichier script (vous pouvez illustrer votre propos avec la première ligne d’un script bash, par exemple) (0.5 point)
* Makefile :
* À quoi sert-il ? (0.5 point)
* Comment est-il organisé ? (0.5 point)
* Comment l’utilise-t-on ? (0.5 point)
* Comment désigne-t-on :
* Le répertoire courant ? (0.5 point)
* Le répertoire parent ? (0.5 point)
* En Unix, quel **symbole** est utilisé pour séparer le nom d’un dossier et son contenu ? (0.5 point)

On vous fournit le programme C **incomplet** suivant :

|  |  |
| --- | --- |
| 01 | #include <stdlib.h> |
| 02 | #include <stdio.h> |
| 03 | #include <unistd.h> |
| 04 | #include <string.h> |
| 05 |  |
| 06 | int mon\_nb\_sec = 0; |
| 07 | char\* ma\_cmd = NULL; |
| 08 | char\*\* ma\_ligne\_de\_cmd = NULL; |
| 09 |  |
| 10 | void analyser\_ligne\_cmd(int argc, char\*\* argv){ |
| 11 | int i=1; |
| 12 | while(i < argc){ |
| 13 | if(strcmp(argv[i], "-d") == 0){ |
| 14 | i++; |
| 15 | mon\_nb\_sec = atoi(argv[i]); |
| 16 | i++; |
| 17 | }else if(strcmp(argv[i], "-e") == 0){ |
| 18 | i++; |
| 19 | ma\_cmd = argv[i]; |
| 20 | i++; |
| 21 | }else if(strcmp(argv[i], "-l") == 0){ |
| 22 | ma\_ligne\_de\_cmd = argv+i+2; |
| 23 | i = argc; |
| 24 | return; |
| 25 | }else{ |
| 26 | printf("erreur\n"); |
| 27 | exit(-1); |
| 28 | } |
| 29 | } |
| 30 | } |
| 31 |  |
| 32 | int main(int argc, char\*\* argv){ |
| 33 | analyser\_ligne\_cmd(argc, argv); |
| 34 | if(ma\_cmd != NULL && ma\_ligne\_de\_cmd != NULL){ |
| 35 |  |
| 36 |  |
| 37 | /\* À COMPLETER (cf. question 13) \*/ |
| 38 |  |
| 39 |  |
| 40 |  |
| 41 |  |
| 42 |  |
| 43 |  |
| 44 | }else{ |
| 45 | printf("Fournissez un exécutable avec l'option -e et une ligne de commande avec l'option -l\n"); |
| 46 | } |
| 47 | return 0; |
| 48 | } |

* À quoi servent les lignes 1 à 4 ? (1 point)
* À quoi servent les paramètres de la fonction main ? (1.5 point)
* Expliquer les lignes 27 et 47. (2 points)
* Dessiner le diagramme d’activités de la fonction analyser\_ligne\_cmd. (3 points)
* Qu’est-ce qu’un pid ? (1 point)
* Compléter le programme C ci-dessus de manière à ce qu’il réaliser le diagramme d’activités ci-dessous. (4, 5 points)

