

QUESTIONNEMENT

Le candidat devra composer sur ce document exclusivement. Si la place allouée à certaines réponses est insuffisante, il pourra utiliser la page blanche située en vis à vis. Ne pas ajouter de feuilles volantes à ce document.

TABLE DES MATIERES DU DOCUMENT REPONSE

TABLE DES MATIERES DU DOCUMENT REPONSE	2
Barème prévisionnel donné à titre indicatif:	3
DOCUMENT REPONSE : Etude d'un système et/ou d'un processus technique	4
A) Étude systèmes OS9 : Architecture physique	4
Première partie : Carte CPU 10	5
Question A-1.1	5
Question A-1.2	5
Question A-1.3	5
Question A-1.4	5
Question A-1.5	5
Question A-1.6	6
Question A-1.7	6
Question A-1.8	6
Deuxième partie : Carte RAM.....	7
Question A-2.1	7
Question A-2.2	7
Troisième partie : Carte Ethernet.....	7
Question A-3.1	7
Question A-3.2	8
Question A-3.2	8
Question A-3.3	8
Quatrième partie : Carte communication Série.....	8
Question A-4.1	8
Question A-4.2	8
Question A-4.3	8
Question A-4.4	9
Question A-4.5 (communication série RS232).....	9
Question A-4.6 (communication série RS232).....	9
Question A-4.7 (communication série RS232).....	9
B) Étude du système UC.	10
Première partie : Spécification	10
Question B-1	10
Deuxième partie : Conception	12
Question B-2.1	12
Question B-2.2	12
Question B-2.3	13
Question B-3.1	13
Question B-3.2	13
Question B-4.1	13
Question B-4.2	14
Question B-4.3	14
Question B-4.4	14
C) Étude de la partie réseau.	15
Première partie : Architecture réseau.....	15
Question C-1	15
Question C-2	15
Deuxième partie : communication réseau Unix - Windows	15
Question C-3	16
Question C-4	16
Question C-5	19
Question C-6	19
Question C-7	19
Question C-8	21
Troisième partie : programme demandes manutentionnaires	21

Question C-9 (analyse programme poste manutentionnaire).....	21
Question C-10 (analyse programme poste manutentionnaire).	22
Question C-11 (analyse UML pour un programme en C++ sur Windows prog: manutentionnaire). 22	
Question C-12 (Programmation : recherche si une année est bissextile).....	25
Quatrième partie : Réseau Ethernet.....	25
Question C-13.....	25
Question C-14.....	26
D) Étude de la partie Supervision.	26
Première partie : Programmation Windows en C++.....	28
Question D-1 (UML).....	28
Question D-2 (programmation Windows).....	29
Question D-3 (programmation sous Windows).....	29
Question D-4 (programmation Windows).....	30
Deuxième partie : Programmation Socket sous Windows en C++.....	30
Question D-5 (programmation socket sous Windows).....	30
Question D-6 (programmation socket sous Windows).....	30
Question D-7 (programmation socket sous Windows).....	30
Question D-8 (programmation socket sous Windows).....	31
Question D-9 (programmation socket sous Windows).....	31
Question D-10 (programmation socket sous Windows).....	31

Barème prévisionnel donné à titre indicatif:

1 point: sera attribué à toutes les questions, sauf pour les questions suivantes:

2 points: A-1.4, A-1.7, B-2.1, B-4.2, B-4.3, C-7, C-9, C-15, D-2, D-3

3 points: A-1.8, D-1, A-4.7

5 points: B-1

6 points: C-5, C-8, C-12

DOCUMENT REPONSE : Etude d'un système et/ou d'un processus technique

A) Étude systèmes OS9 : Architecture physique

Remarque : Le système UC OS9 Mourepiane est en fait composé de deux systèmes :

- UC OS9 (maître) : unité de contrôle d'accès.
- UD OS9 (esclave) : unité déportée d'application.

Description du système (RACK UD-OS9) :

COMPOSANTS MATERIELS :

Le matériel se présente sous la forme d'un rack 19 pouces 3U.

L'unité centrale GMI-CPU10 (68030) a été conçue autour du Bus G96.

Cette démarche permet d'utiliser toutes les cartes au bus G64/G96, qui respectent cette norme.

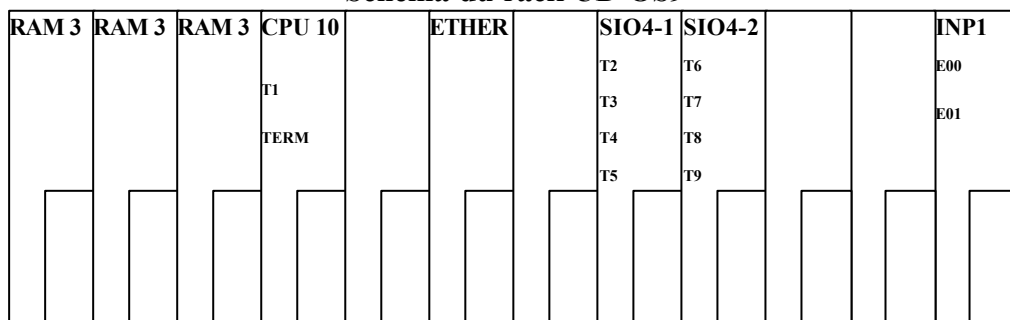
- Mémoire cachée interne
- MMU intégré
- Demande de DMA gérée par le Bus
- Interruptions auto-vectorisées,
- Interruptions vectorisées (191) avec protocole de lecture du vecteur,
- Zone VPA pour les périphériques en mode synchrone,
- Zone VPA pour les périphériques en mode asynchrone,
- Mémoire centrale extensible à 14 Moctets

L'intégration d'un chien de garde, de ressources internes (RAM, EPROM, Horloge et 2 liaisons de communication) sur cette carte permet de réaliser des applications industrielles.

Les cartes suivantes sont implantées :

- 1 Carte GMI-CPU10 avec processeur 68030
- 3 Cartes RAM 3
- 1 Carte Ethernet
- 2 Cartes de communication série (SIO4-1, SIO4-2)
- 1 Carte TOR (INP1)

Schéma du rack UD OS9



COMPOSANTS LOGICIELS :

La machine est livrée avec le système d'exploitation OS9 V2.4 et le progiciel TCP/IP installés dans des EPROMS spécifiques.

Dans le répertoire des descripteurs (/h0/cmds/bootobjs) figurent les descripteurs des liaisons séries de la carte CPU (Term, t1) et des cartes SIO4 (t2, t3, t4, t5, t6, t7, t8, t9).

Première partie : Carte CPU 10

La carte CPU10 est équipée d'un microprocesseur MC68030

Question A-1.1

Quelle est la capacité d'adressage physique du 68030 ?

Quelle est la capacité d'adressage maximale de la carte GMI-CPU10 ?

Question A-1.2

Le bus G96 ne peut gérer que 24 lignes d'adresses (A0-A23). Comment le fabricant de la carte CPU10 a-t-il procédé pour gérer 32 Mégaoctets ?

Question A-1.3

Quelles sont les positions des cavaliers permettant d'utiliser des EPROM de 512K, configurer la carte CPU ?

J18	J20	J21	J22	J24
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

La carte GMI-CPU10 permet de choisir des interruptions vectorisées et/ou auto vectorisées

Question A-1.4

Présenter sous forme de séquences le déroulement du traitement d'une interruption logicielle ?

Question A-1.5

Pour accéder aux différents octets, le 68030 possède deux signaux SIZ0 et SIZ1

SIZ1	SIZ0	Nombre d'octets
1	1	3
1	0	2
0	1	1
0	0	4

Donner les équations de la Pal 10 permettant de sélectionner chaque octet (ne pas tenir compte de FC0, FC1, FC2) sachant que :

- UUD2 valide D24-D31
- UMD2 valide D16-D23
- LMD2 valide D8-D15
- LLD2 valide D0-D7

UUD2 =

UMD2 =

LMD2 =

LLD2 =

Question A-1.6

La carte GMI-CPU10 utilise des PALs pour réaliser le décodage d'adresses. En vous servant du découpage mémoire décrit en Annexe GMI-CPU10 A-1 et du schéma de la carte GMI-CPU10 A-1-2, complétez (avec des 0, 1 ou x) les tables ci-dessous afin d'assurer le décodage des zones RAM.

Zone RAM

A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00	

Question A-1.7

Complétez (avec des 0, 1 ou x) les tables ci-dessous afin d'assurer le décodage des périphériques suivants:

- MC68681 Registre du circuit
- Watch-Dog D0=1 → Déclenchement

MC68681 Registre du circuit

A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00	

Watch-Dog D0=1 → Déclenchement

A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00	

Dans le cadre du décodage d'adresse des zones déterminées par le constructeur (voir Annexe GMI-CPU10 A-1) au travers de la Pal 10

Question A-1.8

Donnez les équations utilisant le minimum de lignes pour décoder chaque zone (uniquement les lignes d'adresse).

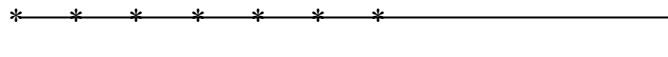
	A24	A23	A22	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
VMA(Réservé)																									
VMA																									
RAMEPROM																									
EPPROM																									
VPA Asynch																									
VPA Synch																									
Zone redond																									
VPA Ext Asy																									
VPA Ext Syn																									
VMA																									
Ext RAM																									
RAM																									

La Carte Ethernet GMI ETHER (Annexe A-3 : GMI ETHER) est installée à l'offset \$1B8, comment configurer les cavaliers de J1 en sachant que l'adresse de base est VPA+0

Question A-3.2

Configurer les cavaliers de J1

A9	A8	A7	A6	A5	A4	A3
*	*	*	*	*	*	*



Question A-3.2

Quelle est l'adresse physique du premier registre de la carte ethernet ?

Question A-3.3

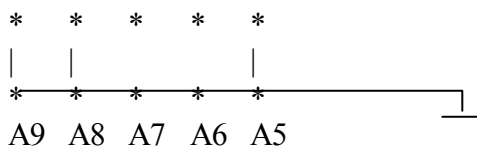
Quelles sont les couches du modèle OSI correspondant aux fonctionnalités d'une carte interface réseau?

Quatrième partie : Carte communication Série

La carte de communication série est dans la zone VPA (Annexe A- 4 : CARTE GMI SIO4)

Question A-4.1

Quelle est l'adresse physique du premier registre de la carte si les cavaliers sont configurés de la manière suivante?



Question A-4.2

Quelle est l'adresse du DUART1?

Question A-4.3

Citer plusieurs protocoles de communication série ainsi que leur type (matériel, logiciel)

Le tableau suivant décrit les caractéristiques des liaisons séries

Port	Interface	Format	Type de liaison
/term			
/t1			
/t2	RS485	9600,8,1,sans	RTS-CTS
/t3	RS485	9600,8,1,sans	RTS-CTS
/t4	Modem	9600,8,1,sans	XON-XOFF
/t5	Modem	9600,8,1,sans	XON-XOFF
/t6		9600,8,1,sans	XON-XOFF
/t7		9600,8,1,sans	XON-XOFF
/t8		9600,8,1,sans	XON-XOFF
/t9	Modem	4800,8,1,sans	XON-XOFF

Les ports /t2 et /t3 ont une interface RS485

Question A-4.4

Quels sont les critères qui permettent de choisir entre les liaisons RS485, RS232 et une liaison par modem ?

Question A-4.5 (communication série RS232)

Précisez la nature des signaux échangés (tension). Tracez un chronogramme qui représente uniquement l'évolution de la ligne d'émission lors de l'envoi d'un caractère (par exemple 0x31).

Question A-4.6 (communication série RS232)

Sur une DB25 la voie 1 est associée à la masse châssis. A quoi sert-elle? Est-elle identique à la référence signal (voie 7) ? Comment la câble-t-on?

Question A-4.7 (communication série RS232)

Expliquez le rôle des protocoles matériels et logiciels (par exemple RTS-CTS et XON-XOFF).

Deuxième partie : Conception

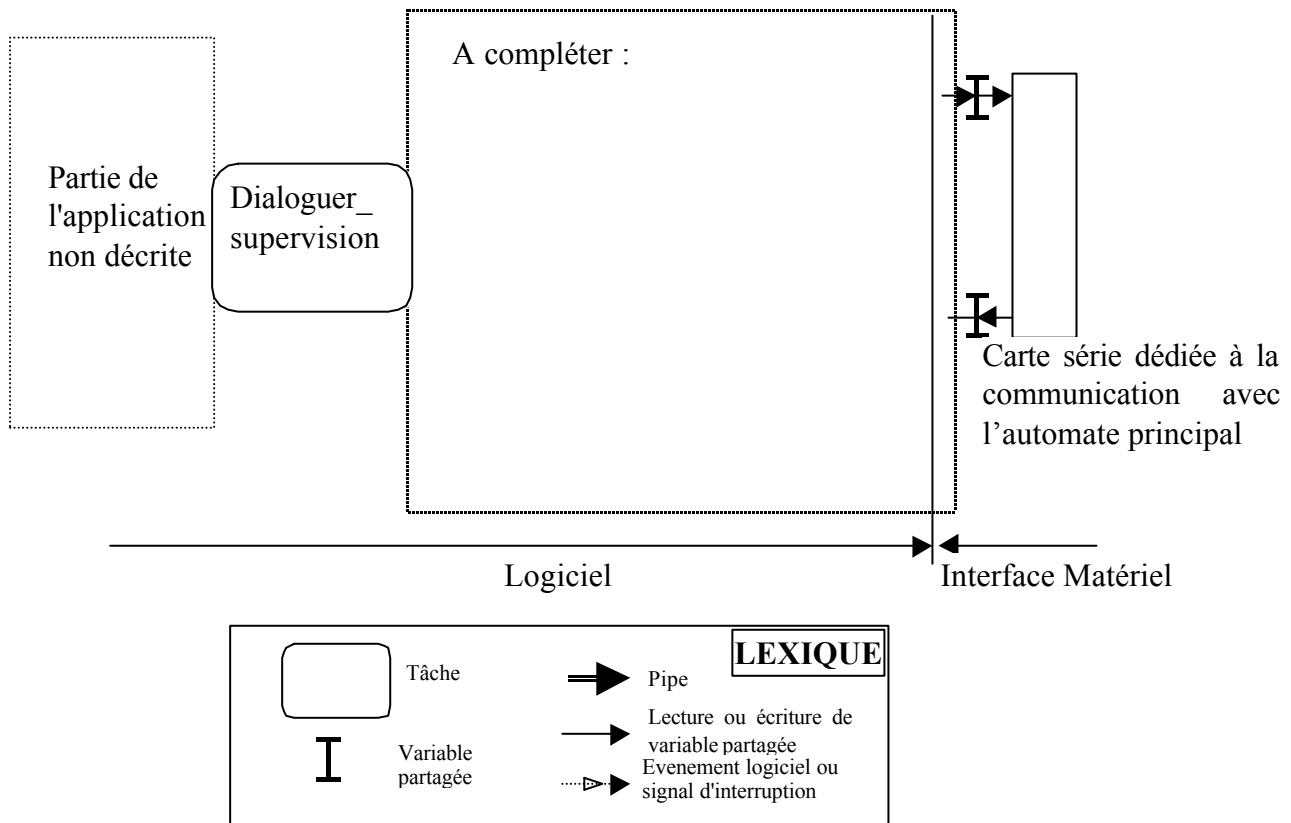
Le système d'exploitation installé sur le système UC OS9 est OS9 68K (S.E. multitâches, temps réel - cf Annexe B-2 et Annexe B-2_1)

Les choix technologiques faits par la société qui a développé l'application, en langage C, sont les suivants :

- La gestion de la représentation des informations Entrées/Sorties T.O.R échangées entre l'automate principal et le système UC OS9 est faite au niveau du bit (l'état d'une Entrée T.O.R. est codé sur 1 bit, idem pour une Sortie T.O.R.)
- Le flot de données «ordres_pilot_autom» est transféré entre deux tâches sans filiation par un pipe nommé.
- Deux tâches partagent une zone de mémoire commune (concept de datamodule sous OS9 68K) qui matérialise la zone de stockage «etats_capteurs» suivant un modèle «producteur/consommateur».
- La tâche «producteur» est une tâche rythmée.

Question B-2.1

En cohérence avec la spécification que vous avez proposée précédemment, compléter le schéma d'architecture partiel de la réalisation logicielle fourni ci-dessous :



Question B-2.2

Justifier le choix technologique de la représentation "au niveau du bit" des informations Entrées/Sorties T.O.R échangées entre l'automate principal et le système UC informatique.

Question B-2.3

En cohérence avec les choix précisés ci-dessus; en langage C, quelle représentation des données proposez-vous pour la zone de stockage "états capteurs».

Question B-3.1

Justifier le choix technologique d'un pipe nommé.

Question B-3.2

Proposer, une fonction `creat_pipe(nom_pipe, ...)` écrite en langage C, qui assure la création d'un pipe en cohérence avec l'application.

Question B-4.1

Quelle(s) solution(s) proposez-vous afin de fiabiliser l'échange d'informations à travers la zone de mémoire partagée entre la tâche «producteur» et la tâche «consommateur»

Identifier les concepts existants sous OS9 68 K permettant de mettre en œuvre ce fonctionnement.

Question B-4.2

Ecrire, en langage C, la fonction "créer_datamodule(...)" conforme à la description suivante :

```
/* nom      : creer_datamodule
   fonction : créer un data module (sur lequel d'autres tâches pourront se lier via modlink() )
   entrées  : nom du data module
             : taille de la zone de données
   retourne : pointeur sur corps du data module (début de l'espace réservé aux données) */
```

Question B-4.3

En cohérence avec la description ci-dessus:

Proposer les séquences d'initialisation qui permettront à la tâche «consommateur» de lire, de façon fiable, l'information contenue dans la zone de mémoire commune via l'usage d'une fonction "lit_datamodule(...)".

Préciser les paramètres à passer à la fonction "lit_datamodule(...)".

Question B-4.4

Ecrire, en langage C, la fonction "lit_datamodule(...)"

C) Étude de la partie réseau.

Première partie : Architecture réseau.

Question C-1

Identifier et décrire les caractéristiques principales des bus de terrain :

- Mis en œuvre au sein de cette application, s'il(s) existe(nt)?
- Que vous connaissez ?

Question C-2

Quelle est la fonction d'un routeur?

Identifier, s'il(s) existe(nt) le(s) routeur(s) présent(s) sur le schéma d'architecture fourni en Annexe générale G-1.

Dans le cadre des caractéristiques et composants des routeurs, préciser :

- des interfaces LAN et WAN
- des protocoles réseaux
- des protocoles de routage

Deuxième partie : communication réseau Unix - Windows

Afin d'autoriser la sortie de conteneurs, les sociétés de manutention dont les sièges sociaux sont situés hors du site de Mourepiane :

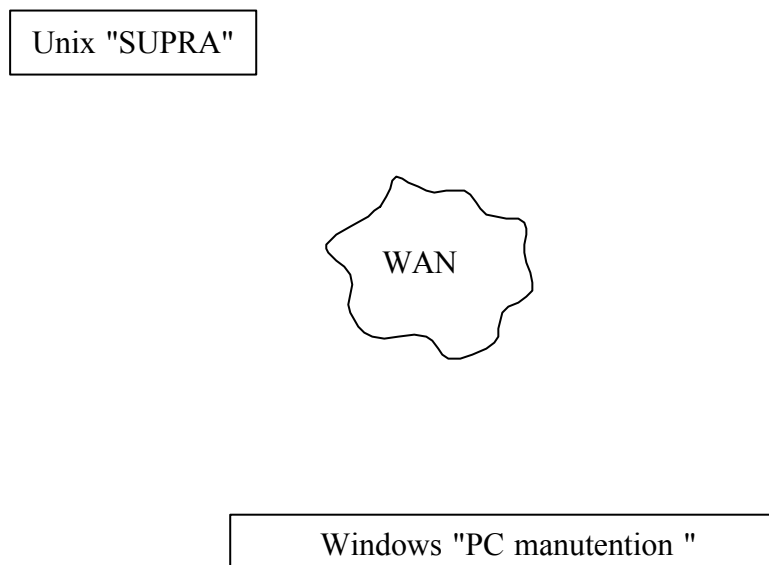
Transmettent, via un PC Windows, vers la machine Unix "SUPRA" du site de Mourepiane , un datagramme contenant :

- ◆ le login de la société de manutention (nom d'identification et password)
- ◆ le type de l'opération demandée:
 - TYPE_LOGIN : uniquement contrôle du login (nom d'identification et password de la société de manutention)
 - TYPE_CONVOI_ENTREE : contrôle du mot de passe + informations complémentaires (optionnelles).
 - TYPE_CONVOI_SORTIE : contrôle du mot de passe + informations complémentaires.
- ◆ Des informations complémentaires :
 - la date et l'heure prévisionnelle du transfert des conteneurs
 - le numéro de badge
 - le numéro d'immatriculation du tracteur PL

- le nombre de colis (0 : tracteur seul, 1..2)
- des informations sur le colis 1 :
 - la nature (0 : conteneur, 1 : autre)
 - le mode (0 : normal, 1 : transit, 2 : douane)
 - le nombre de conteneurs (0..4)
 - la liste des conteneurs (tableau statique)
 - des commentaires additionnels
- des informations sur le colis 2 : (idem colis 1)
- ♦ le crc

Question C-3

Compléter le schéma représentant l'architecture physique du réseau, compatible avec le schéma d'architecture fourni en Annexe générale G-1 permettant aux sociétés de manutention de communiquer avec le PC Unix "SUPRA".



Question C-4

Proposer, en langage C, au niveau de la machine "Unix SUPRA" une solution commentée qui gèrera la communication réseau par datagramme (PC(s) Windows manutention – Unix SUPRA).

Vous disposez d'une «documentation Socket Unix» cf Annexe C-1 + «Windows Sockets» cf Annexe D-3

Ce **serveur** sera :

- Compatible avec le fonctionnement décrit ci-dessus.
- Compatible avec le programme source partiel fourni en Annexe D-1 (fichier manut.cpp).
- Lancé à partir du shell Unix en background.

Hypothèses : (pour les types à manipuler se référer à l'Annexe D-2 fichier types.h)

Vous disposez des primitives suivantes :

```

/* parcourt la BD : si nom du «user» non trouvé alors rend ERR_NOM_INCONNU
   sinon si password associée à nom user NON VALIDE alors rend ERR_PASSWD
   sinon rend OK */

```

```

unsigned short verifie_login(TRAME_CMD_UNIX *trame);

```

```

/* parcourt la BD : si numéro carte non trouvé ou carte invalide
   alors rend ERR_CARTE_NON_VALIDE
   sinon rend OK */

```


Question C-5

On désire réaliser un contrôle de trame à chaque envoi et réception de trame réseau de type «struct Trame_Cmd_UNIX» (voir Annexe D-1, types.h):

Citez quelques méthodes de contrôle de trame et précisez leurs avantages ou inconvénients.

On vous demande de réaliser la fonction suivante *unsigned short calculer_crc (unsigned char *trame, int size)* où le résultat renvoyé est l'addition non signée (modulo 0xFFFF) de tous les octets de la trame de manière à satisfaire l'appel suivant:

```
if(trame.crc == calculer_crc((unsigned char *) trame, sizeof(trame)))
```

Cette fonction est utilisée par la fonction *verifie_crc()*.

Question C-6

On désire lancer automatiquement le processus **serveur** au démarrage de inetd (cf Annexe C-2), fournir des extraits des fichiers de configuration permettant ce fonctionnement

Question C-7

Fournir le code C du serveur :

- "Lancé" par inetd (cf Annexe C-2)
- Capable de gérer, à un instant t, qu'une seule requête issue de société manutentionnaire.

Vous disposez des primitives décrites à la question C-5.

Question C-8

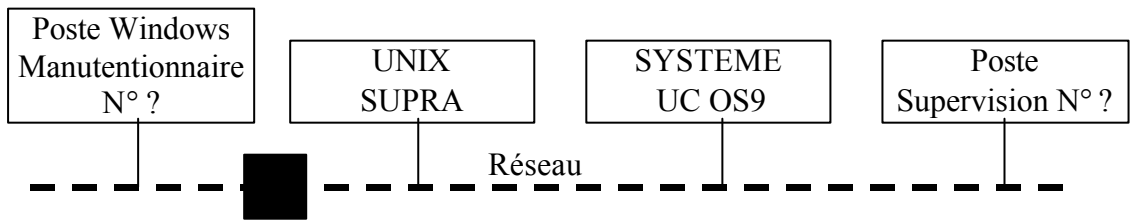
Est-il possible d'écrire un serveur :

- "Lancé" par inetd
- Capable de gérer "simultanément" plusieurs requêtes datagrammes issues de sociétés manutentionnaires .

Dans le cadre d'une réponse affirmative:

- Les fichiers de configuration décrits par vos soins en réponse à la Question C-7 doivent-ils être adaptés.
- Exposer les grandes lignes de votre solution.

Troisième partie : programme demandes manutentionnaires



Question C-9 (analyse programme poste manutentionnaire).

En sachant que le programme est écrit en C++ Borland 3.1 pour Windows (programmation Objet 16 bits, avec des bibliothèques objets fournies), proposez un cycle de développement adapté à ce type de programmation et justifiez votre proposition ?

Question C-10 (analyse programme poste manutentionnaire).

Donnez le rôle d'un atelier de génie logiciel (AGL) dans la conduite d'un projet en informatique industrielle.

Question C-11 (analyse UML pour un programme en C++ sur Windows prog: manutentionnaire).

On vous demande de réaliser uniquement les vues : Use Case View et Logical View ; de l'analyse UML avec les diagrammes nécessaires afin de préciser la phase de «login».

Le cahier des charges imposé par les manutentionnaires et le gestionnaire de service est le suivant :

- Lancement du programme.



- Demande de login

- Teste si ce login est correct quand on valide avec envoi vers le serveur UNIX.
- Stockage pour toute la session du login (nom, passwd) si réponse du serveur OK avec nouveau menu.



- Saisie de convoi voie P.L., ...

- Validation avec envoi vers le serveur UNIX
- Réponse du serveur
- Possibilité de nouvelles saisies, ...

Question C-12 (Programmation : recherche si une année est bissextile).

On vous demande d'écrire en C la fonction *int bissextile(int annee)* qui renvoie 0 (année non bissextile) ou 1 (année bissextile).

Nota: Une année est bissextile si :

- Elle est multiple de 4.
- Sauf si elle est multiple de 100 ; elle l'est quand même si elle est multiple de 400.

Quatrième partie : Réseau Ethernet.

L'environnement TCP/IP a été configuré comme suit:

Dans le fichier /h0/internet/ISP/driver/GMI_ETH1/gmo.a

Broadcast_addr
dc.b 128.6.255.255 broadcast address (internet)
interface_address
dc.b 128.6.1.101 direct address (internet)

Dans le fichier /h0:internet/ISP/ipconfig/ipconfig.a

*Internet configuration
dc.l 0x0001 gateway flag
dc.b 128.6.0.1 default internet destination

Dans le fichier /h0/internet/etc/hosts

127.0.0.1	localhost
128.1.10.126	UC_MAR
128.1.4.83	UC_GARE
128.1.10.121	SUPRAMAR
128.1.2.1	SUPRAFOS
128.6.1.101	UC_MOUREPIANE
128.6.1.102	UD_MOUREPIANE
128.6.1.103	PC_OP1
128.6.1.104	PC_OP2
128.6.1.105	PC_OP3
128.6.1.106	PC_CARTE
128.6.1.107	SUPRA_MOUREPIANE

Question C-13

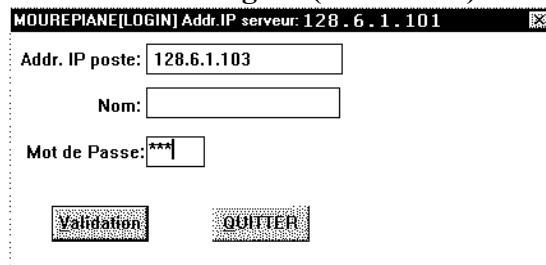
Citer les classes de réseaux ainsi que les plages d'adresses IP existantes au sein du monde IP.

Les principales fenêtres de l'interface homme-machine du superviseur sont les suivantes :

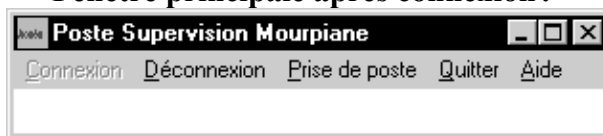
Fenêtre principale au départ :



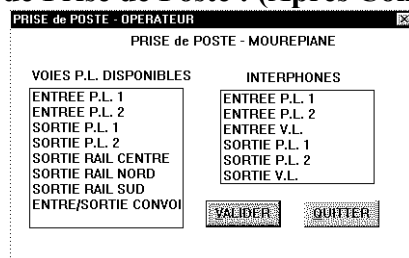
Fenêtre de login : (Connexion)



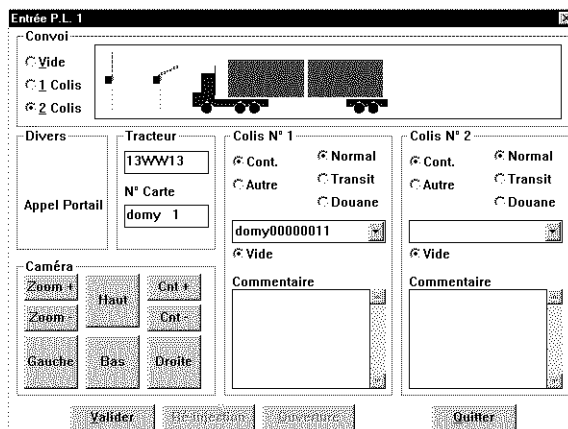
Fenêtre principale après connexion :



Fenêtre de Prise de Poste : (Après Connexion)



Fenêtre d'Entrée P.L. 1



Question D-4 (programmation Windows)

Comment une fenêtre fille peut-elle dialoguer avec sa fenêtre mère (dans les deux sens) ?

Deuxième partie : Programmation Socket sous Windows en C++

Question D-5 (programmation socket sous Windows)

A partir du programme source «superv_pl.cpp» en Annexe D-2 et de l'Annexe D-3 documentation «Windows Sockets», expliquez la méthode *int TFenetrePrincipale: :socket_client()*.

Pourquoi y a t-il 2 sockets?

Précisez le rôle de la fonction *WSAAsyncSelect(...)*?

Question D-6 (programmation socket sous Windows)

Que se passe-t-il quand le serveur implanté sur le système UC_OS9 se déconnecte ?

Question D-7 (programmation socket sous Windows)

Que se passe-t-il quand le serveur implanté sur le système UC_OS9 envoie une trame de commande ?

Question D-8 (programmation socket sous Windows)

En sachant que le programme source a été créé pour le système d'exploitation Windows 3.1 avec un compilateur Borland 3.1 (16 bits); avec les compilateurs et les systèmes d'exploitation (Windows 95,...) plus récents pourrait-on se passer de la fonction *WSAAsyncSelect(...)* et dans ce cas quelle forme aurait le programme.

Question D-9 (programmation socket sous Windows)

En sachant que le programme superviseur est implanté sur un PC avec Windows 3.1 (ce programme a été compilé avec le compilateur Borland C++ 3.1 pour Windows 16 bits), que le programme de réception est implanté sur une machine 68000 avec OS9 68K et un compilateur C++ 32 bits, les trames seront elles directement utilisables sur la machine 68000? Sinon donnez les principes d'une solution.

Question D-10 (programmation socket sous Windows)

Est-il nécessaire de réaliser un contrôle de trame en mode connecté?
