Ecole Nationale Supérieure des Bibliothécaires

Diplôme d'Etudes Supérieures Spécialisées : Informatique Documentaire

Rapport de stage: 1er Juin-30 Septembre ARIST Rhône Alpes

MISE EN PLACE

DU SERVICE TELEMATIQUE

ARISTOTELE

Jean-Pierre Poulet

Année 1987-1988

Directeur de stage: Monsieur Michel Vuaillat, Directeur de L'ARIST

Ecole Nationale Supérieure des Bibliothécaires

Diplome d'Etudes Supérieures Spécialisées : Informatique Documentaire

Rapport de stage: 1er Juin-30 Septembre ARIST Rhône Alpes

MISE EN PLACE

DU SERVICE TELEMATIQUE

ARISTOTELE

an-Pierre Poulet

Année 1987-1988

irecteur de stage:Monsieur Michel Vuaillat, Directeur de L'ARIST

Comme j'étais surtout interessé par l'information en direction des entreprises j'ai tout naturellement demandé à pouvoir faire mon stage dans une ARIST. Je me suis adressé à l'ARIST Rhône Alpes et celle-ci pouvait proposer un stage interessant dans le cadre de la mise en place de son service télématique ARISTOTELE .Les choses se sont donc faites simplement j'ai été accepté pour un stage de quatre mois au sein de l'ARIST Rhône Alpes.

Ce rapport sera organisé autour de trois parties:

_ Qu'est ce que l'ARIST?

_ Mon rôle et mon travail à l'ARIST

_ Mon travail de programmation en Pascal

I QU'EST CE QUE L'ARIST?

L'ARIST ou Agence Régionale d'Information Scientifique et Technique est un service de la Chambre Régionale de Commerce et d'Industrie depuis 1976.

L'accèlèration des progrès de la technique, la mutation industrielle qui s'est produite depuis le premier choc pétrolier ont conduit les Chambres de Commerce et d'Industrie à crèer de nouveaux types de services à dominante technologique. D'abord la mise en place dans chaque Chambre de Commerce et d'Industrie de

Conseillers d'entreprise appelés Assitants Techniques à l'Industrie(ATI). Ils devaient convaincre les chefs d'entreprise d'abandonner les méthodes désuètes, pour des méthodes nouvelles ayant fait leurs preuves.

Puis en 1976 à l'initiative du Ministère de l'Industrie qui était conscient du retard pris par notre pays pour la diffusion de l'information, les Chambres de Commerce et d'Industrie ont crèé dans chacune des régions de France (il y en a 22) une AGENCE REGIONALE D'INFORMATION SCIENTIFIQUE ET TECHNIQUE - ARIST. Ce service a pour mission de contribuer au développement technologique régional et à la promotion de l'innovation industrielle dans les petites et moyennes entreprises par une meilleure diffusion de l'information scientifique et technique.

Les ARIST se trouvent ainsi aux avant-postes des actions des Chambres de Commerce et d'Industrie en faveur des petites et moyennes entreprises industrielles.

Le rôle d'une ARIST est de fournir au chef d'entreprise l'information scientifique et technique nécessaire à la résolution de ses problèmes et destinée à conforter sa réflexion avant la prise de décision.

Comme le chef d'entreprise n'est pas très friand d'informations écrites. Il est donc nécessaire de crèer une méthodologie de travail spécifique, qui rende l'information accessible à ce chef d'entreprise débordé.

L'information devra être concise. l'ARIST recueille des informations à de multiples sources, elle ne devra pas se limiter à fournir une pile de documents de plusieurs centimètres d'épaisseur: elle devra en faire la collation et la synthèse.

Les chef d'entreprise parlent un langage qui est différent de celui du chercheur de laboratoire. Ils n'aiment pas beaucoup les langues étrangères. L'ARIST devra traduire et utiliser un langage clair, compréhensible par ses clients.

L'information transmise aura été recueillie rapidement. L'entreprise ne considère l'information comme pertinente que si son exploitation peut être immédiate.

Enfin la finalité économique des décisions qu'aura à prendre le chef d'entreprise devra être présente tout au long de la recherche de l'information et de la rédaction de la réponse.

La petite et la moyennne entreprise sont à la recherche d'une information personnalisée et l'ARIST doit leur fournir du "sur mesure".

L'ARIST analyse ses interventions selon quatre niveaux qui correspondent à une valeur ajoutée de plus en plus importante.

AGENCE REGIONALE D'INFORMATION SCIENTIFIQUE ET TECHNIQUE

4 NIVEAUX D'INTERVENTION	MOYENS MIS EN OEUVRE
1er NIVEAU Réponse aux questions "QUI FAIT QUOI ? , OU ? "	. fichiers sectoriels - centres techni. ques, annuaires professionnels, répertoires, ... fichiers informatisés spécialisés (normes, labos, etc...)
2ème NIVEAU Fourniture de documents : - interrogation des fichiers informatisés scientifiques et techniques, de brevets, de transferts de technologie. - recherche des documents primaires signalés par les listings	. fichiers informatisés scientifiques et techniques (des principaux serveurs mondiaux...), de brevets et de transferts de technologie. . centres de documentation sectoriels et de propriété industrielle.
ème NIVEAU laboration de dossiers avec valeur joutée état de la technique situation technico-économique point sur la réglementation, les normes e sont des recherches d'informations uivies de synthèses, avec apport 'informations économiques, évaluation es tendances...	. même sources qu'au 2ème NIVEAU en plus : . réseau de spécialistes extérieurs, de la recherche publique ...
ème NIVEAU ervice d'aide à la définition de la tratégie industrielle ou innovatrice e l'entreprise.	. approche méthodologique . utilisation de fichiers, appel à des Cabinets Conseils ou des Sociétés de services privés.

Le premier niveau consiste à répondre à des questions simples du genre "qui fait quoi, où ?"

Cette recherche nécessite surtout de consulter des annuaires professionnels, des répertoires et l'accès à des fichiers informatisés spécifiques, tels Labinfo qui donne la liste des labos du CNRS, Noriane la liste des normes françaises,etc...

Dans le second niveau, nous classerons les interrogations des bases et banques de données scientifiques, techniques, économiques, de transfert de technologie et leurs compléments:

_ l'analyse des résultats et l'acquisition des documents primaires signalés par les listings.

L'interrogation est un travail de spécialiste, la recherche de documents primaires est souvent difficile. Ce second niveau commence à introduire dans la prestation une valeur ajoutée certaine.

Au troisième niveau, nous placerons les prestations qui constituent presque toujours plus de 50% de l'activité d'une ARIST: élaboration de dossiers à forte valeur ajoutée, il s'agit:

_ d'états de la technique, c'est le point le plus récent obtenu grâce à la littérature et les brevets.

_ de sondages d'antériorité en propriété industrielle, préalablement au dépôt d'un brevet.

_ d'études technico-économiques, qui donnent la photographie de l'environnement économique.

Les informations recueillies sont recoupées. Une synthèse est réalisée, l'information économique prise en compte. Des idées force sont dégagées. Il y a donc là un important travail de valorisation.

Le quatrième niveau se rapproche beaucoup de l'activité de conseil. Il s'agit d'aider l'entreprise dans ses travaux préparatoires à la définition d'une stratègie dans le domaine de l'innovation ou du développement économique, industriel ou commercial.

L'approche méthodologique réalisée par l'ARIST est souvent poursuivie en faisant appel à des cabinets conseils ou des sociètés de services spécialisés du secteur privé.Les ARIST sont financées d'une part sur le budget des Chambres de Commerce et d'Industrie, budget alimenté par un impôt payé par les entreprises, d'autre part par la rémunèration des travaux comportant de la valeur ajoutée. Les travaux comportant de la valeur ajoutée ou des travaux sous-traités (interrogation de bases de données par exemple) sont facturés aux entreprises pour une part parfois importante de leur coût de revient.

Ces deux financements peuvent au premier abord sembler contradictoires mais en fait ce n'est pas le cas. Au cours de la décennie 1980 qui est celle de l'expansion de l'informatique documentaire - liant automatiquement la notion de coût à celle de l'acquisition de l'information - il n'est pas concevable de fournir gratuitement une information valorisée. Une telle informaton doit être considèrée comme une matière première du développement industriel et de l'innovation.

Une fourniture gratuite aurait deux inconvénients:

_ le sérieux du travail qui a précèdé la fourniture "gratuite" pourrait être mis en doute.

_ les entreprises de faible taille ne prendraient pas conscience de l'évolution en cours qui attribue un coût à l'information.

Les grandes entreprises, grandes soit par leur taille ou par leurs performances mondiales ont depuis longtemps déjà accepté de payer l'information à son juste prix. Il n'en est pas encore de même pour toutes les petites et moyennes entreprises qui représentent la presque totalité des clients des ARIST.

Les ARIST sont aujourd'hui au nombre de 22, autant qu'il y a de régions (voir la carte).

LES AGENCES REGIONALES D'INFORMATION SCIENTIFIQUE ET TECHNIQUE

On estime la taille minimale d'efficacité à 5 personnes.

_ deux ingénieurs qui traitent les dossiers d'entreprises

_ Un ingénieur qui analyse les prévisions technologiques

_ Un technicien affecté aux transferts de technologie

_ un documentaliste et un secrétariat.

Les ARIST de France se sont constituées en réseau afin d'accroitre leur efficacité.

L'ARIST RHONE ALPES

L'ARIST Rhône Alpes compte aujourd'hui 7 personnes, 4 ingénieurs, 1 technicien supérieur, 2 secrétaires.

Elle a facturé 168 dossiers en 1987. Le montant des facturations s'est accru de 25 % par rapport à 1986 à effectif égal, traduisant un accroissement des prestations à valeur ajoutée. Ne sont pas comptabilisés les nombreux rendez-vous et travaux qui n'ont pas donné lieu à facturation, ni les devis sans suite. L'ARIST a répondu comme chaque année à plusieurs milliers de demandes téléphoniques.

Elle a aussi réalisé pour le compte de la CRCI et du Conseil Régional "Le guide du développement technologique de l'entreprise en Rhône Alpes" distribué à 20.000 exemplaires.

L'ARIST mène aussi une politique de promotion et d'explication de son travail auprès des entreprises à travers la participation à des expositions, des foires, des colloques, des

visites aux grandes écoles et aux entreprises, des permanences dans les différentes Chambres de Commerce et d'Industrie.

Enfin cette année l'ANVAR en application des orientations du Comité Interministèriel du 8 Juillet 1987 s'est associée avec les Chambres de Commerce et d'Industire pour lancer une opération nationale "coup de poing", " Les 1000 dossiers d'Information des Entreprises sur les Technologies". Son but est de sensibiliser les PMI de moins de 500 personnes à l'apport scientifique, technologique et économique. Cette opération s'appuie sur le travail de l'ARIST. L'ARIST Rhône Alpes attend une centaine de dossiers dans le cadre de cette opération.

Le contenu du dossier peut porter sur les divers aspects de l'information stratègique utile à l'entreprise:

_ le niveau actuel atteind par la technologie

_ la liberté d'exploitation (les brevets gènants)

_ quel est le marché actuel et sa tendance?

_ quelle est la concurrence ?

etc..

Chaque dossier dont la valeur réelle a été fixée forfaitairement pour cette opération à 10.000 F HT sera payé 2500 F seulement par l'entrpeise , ANVAR et CRCI prenant à leur charge la différence.

Le système informatique de 1'ARIST

La Chambre Régionale de Commerce et d'Industrie Rhône Alpes est équipée d'un IBM 36 minicompact de 1024 ko de RAM et 210 megas octets de mémoire sur disque ,28 postes connectables.
Six terminaux sont connectés:

_ 2 PC XT émulation 5250

_ 2 claviers écrans monochromes 3196

_ 1 clavier écran monochrome 3197

_ 1 clavier écran couleur 3197

Sont connexées aussi :

_ 2 imprimantes matricielles 4202

_ 1 imprimante transfert thermique 5202

II MON ROLE A L'ARIST

En plus des missions que nous avons décrites dans la première partie l'ARIST anime un service télématique qui a été crèé en 1985 à l'initiative de la Fondation Scientifique de Lyon sous l'impulsion du Conseil Régional. Ce service avait été crèé par le CRITT G3F et pris en charge par l'ARIST pour sa maintenance. Ce service appelé ARISTOTELE code 36 15 ISTRA était chargé sur le serveur G.CAM. Un logiciel de gestion aujourd'hui ancien offrant peu de possibilités et mauvais sur le plan ergonomique a eu pour conséquence le dépérissement du service car il était très difficile avec ce logiciel de mettre les informations à jour. En 1988 la CRCI, l'ARIST et La Fondation Scientifique de Lyon ont décidé de rajeunir ce service.

Ils ont décidé de changer de serveur. Le service passe du G.CAM au SUNIST et sera gèré par un logiciel beaucoup plus fonctionnel. Dans le même ordre d'idée il était nécessaire de vérifier l'ensemble des informations qui seraient intègrées dans le service et de redéfinir ce qui devrait se trouver dans le service. C'est pour participer à l'ensemble de ces taches que j'ai été pris en stage au sein du service de l'ARIST.

QU'EST CE QU'ARISTOTELE ?

ARISTOTELE est une application vidéotex dont la mission est de décrire les principaux acteurs du transfert de technologie dans la région Rhône Alpes. Cela signifie recenser toutes les associations participant à ce transfert, les centres techniques, les CRITT, les pôles de compétences, les groupements scientifiques, les laboratoires, les universités, les grandes écoles,IUT, Juniors entreprises, les centres de documentation, les producteurs de bases de données, les institutions susceptible de financer ce transferts de technologie ainsi que tous les acteurs organismes publics ,consulaires ou organismes patronaux acteurs de ce transfert de technologie.

Exemple d'une fiche de renseignements fournie par ARISTOTELE sur un organisme (nous avons choisi la fiche de l'ARIST elle même)

SIGLE : ARIST RA

NOM DE L'ORGANISME :Agence Régionale d'Information Scientifique et Technique Rhône Alpes

ADRESSE :

Quai Achille Lignon

LYON

69459 LYON CEDEX 06

Téléphone: 78 89 29 29

Telex : 900677 CRCIROL

CONTACTS :

Mr Vuaillat Michel

Directeur

Poste 140

Mme DESCHARMES Sylvianne

Ingénieur

Poste 142

DOMAINE :

Pluridisciplinaire

TYPES :

Relais et appuis des entreprises;Chambre de Commerce et d'Industrie.

MOTS CLES :

IST ; BASE DE DONNEES ; PROPRIETE INDUSTRIELLE ; TRANSFERT TECHNOLOGIE ;

COMMENTAIRE :

Pour toute entreprise, la bonne information, livrée au bon moment, donne aujourd'hui la clé de l'efficacité. Les ARIST ont pour vocation de répondre de la façon la plus performante à la demande en information scientifique et économique des entreprises.

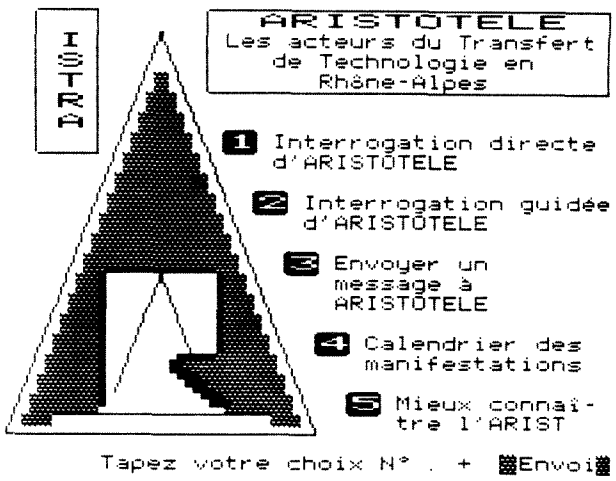
Organisation du service ARISTOTELE

Le service pourra être consulté de deux façons différentes.

_ une consultation guidée sur le mode arborescent.

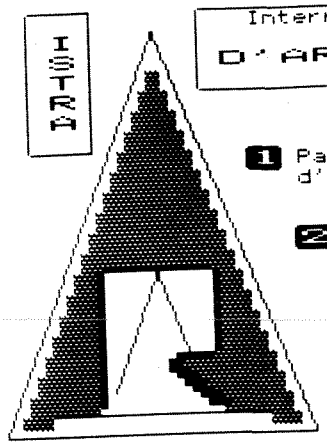
_ une consultation libre grace à une grille de saisie.

Dans la consultation guidée on peut interroger soit par type d'organisme soit par secteur d'activité.



ISTRA

Interrogation guidée
D'ARISTOTELE



1 Par types d'organismes

2 Par secteurs d'activité

Tapez votre choix

N° . + Envoi

ARISTOTELE
TABLE DES ORGANISMES

1 Pôles de compétences
2 Enseignement Supérieur
3 Information et documentation
4 Financement
5 Institutions et organismes publics
6 Relais et appuis des entreprises

Votre choix + ENVOI
Revenir à la sélection ... RETOUR
→

ARISTOTELE
INFORMATION ET DOCUMENTATION

1 Centres de documentation et bibliothèques
2 Presse régionale
3 Producteurs de bases de données

Vous avez 5 réponse(s)

Votre choix
Revenir à la sélection
→ 1

ARISTOTELE
TABLE DES SECTEURS D'ACTIVITE

1 Agriculture Agroalimentaire Agronomie
2 Automat. Robot. Informatique indust.
3 Bâtiment Travaux-publics Urbanisme
4 Chimie parach. Traitements de surf.
5 Electric. Electron. Physique de base
6 Gestion Normalisation Qualité
7 Emballage Conditionnement
8 Energie Environnement Pollution
9 Informat. de Gestion Télécom. Télémat
10 Matériaux Métallurgie Sidérurgie
11 Mécanique Travail des matériaux
12 Mesure Test Contrôle
13 Santé Médecine Pharmacie Sécurité
14 Textile Habillement
15 Tansportss Manutention Stockage
16 Pluridisciplinaire

Votre hoix + ENVOI
Revenir à la sélection ... RETOUR

LISTE DES ORGANISMES — ARISTOTELE

1 SIGLE : ADEMAP
NOM : Agence pour le Développement de la Mise en oeuvre des Matières Plastiques

2 SIGLE : ADERLY
NOM : Association pour le Développement Economique de la Région Lyonnaise

3 SIGLE : ADERME
NOM : Association pour le Développement de l'Enseignement et la Recherche en Microonde s et

Document complet N° + ENVOI
Revenir à la sélection SOMMAIRE

FICHE COMPLETE — ARISTOTELE

SIGLE : ADEMAP

NOM DE L'ORGANISME :
Agence pour le Développement de la Mise en oeuvre des Matières Plastiques

ADRESSE :
181 avenue Jean Jaurès
LYON
69007 LYON
Téléphone : 78 61 26 78

PAGE 1/3 — DOC 1/15
Liste des organismes RETOUR
Autre organisme N° + ENVOI
Revenir à la sélection SOMMAIRE

FICHE COMPLETE — ARISTOTELE

CONTACTS :
Mr GENILLON
Mme CATHAUD

SECTEUR D'ACTIVITE :
Matériaux Métallurgie Sidérurgie

TYPE D'ORGANISME :
Assistance technique et technologique des entreprises ; Associations ; CRITT ; Relais et appuis des entreprises ; Conseillers technologiques ;

PAGE 2/3 — DOC 1/15
Liste des organismes * RETOUR
Revenir à la sélection SOMMAIRE

→

FICHE COMPLETE — ARISTOTELE

MOTS CLES :
PLASTIQUE ; COMPOSITE ;

COMMENTAIRE :
L'ADEMAP met à la disposition des industriels de la région des moyens d'essais et de contrôle dans le domaine des matières plastiques.
L'ADEMAP compte deux établissements, l'un à LYON l'autre à Saint-Etienne.

PAGE 2/3 — DOC 1/15
Autre organisme N° + ENVOI
Liste des organismes * RETOUR
Revenir à la sélection SOMMAIRE

→

Dans la consultation libre une grille de saisie sera proposée qui permettra d'interroger par sigle, titre ou mot du titre, département, secteur d'activité, type d'organisme et mots clés reliés implicitement par l'opérateur booléen OU. L'appel d'écrans d'aide est possible pour chaque rubrique.

ARISTOTELE

Nom ou Sigle

Département

Secteur d'activité

Type d'organisme

Mots clés

Changement de critère SUITE / RETOUR
Voir les réponses ENVOI
Guide du critère GUIDE
Informations générales * GUIDE

ARISTOTELE
GUIDE NOM OU SIGLE

Vous pouvez indiquer au choix :

- l'INTITULE exact
- les TERMES les plus SIGNIFICATIFS de l'intitulé
- le SIGLE

EXEMPLES :

- FACULTE DE PHARMACIE
- AGENCE DEVELOPPEMENT PLASTIQUE
- ADEMAP

Votre choix + ENVOI

ARISTOTELE
GUIDE MOTS-CLES

Tapez un ou plusieurs mots séparés par un espace

Vous obtiendrez toutes les fiches qui contiennent dans leurs mots-clés au moins l'un des termes indiqués

Vous pouvez taper le début des mots

ELECTR appelera ELECTRICITE
ELECTRONIQUE etc.

Vous pouvez si vous le souhaitez consulter la liste des mots-clés en tapant SUITE

Votre choix + ENVOI

CALENDRIER DES MANIFESTATIONS

1 PERIODE : 16-09-88 26-09-88
TITRE : SALON INDUSTRIEL

2 PERIODE : 16-09-88 26-09-88
TITRE : SALON DE L'INVENTION

3 PERIODE : 16-09-88 26-09-88
TITRE : SALON DE LA GASTRONOMIE

4 PERIODE : 22-09-88
TITRE : LE CONTRAT D'AGENCE COMMERCIALE

Vous avez 16 réponses

Document complet N° + ENVOI
Revenir au menu SOMMAIRE

→

LEGENDE DES MANIFESTATIONS

PERIODE : 22-09-88

TITRE : LE CONTRAT D'AGENCE COMMERCIALE A L'ETRANGER

LIEU : CCI GRENOBLE 38

RESUME : LA RÉDACTION DU CONTRAT D'AGENT COMMERCIAL (LES PRINCIPALES CLAUSES, LES PIÈGES À ÉVITER) L'IMPACT DE LA DIRECTIVE EUROPÉENNE

CONTACT : MME ODILE ARNOULD

PAGE 1/2 nnn 4/16

Liste des documents RETOUR
Autre document N° + ENVOI
Revenir au menu SOMMAIRE

→

LEGENDE DES MANIFESTATIONS

TELEPHONE : 76472036

PAGE 2/2 nnn 4/16

Autre document N° + ENVOI
Liste des documents ↕ RETOUR
Revenir au menu SOMMAIRE

Il sera aussi possible d'envoyer un message à ARISTOTELE ce qui permettra à n'importe quel utilisateur de faire une demande particulière à l'ARIST producteur de l'application. La réponse du producteur se fera par des voies non-télématiques.

Un dernier module donnera un calendrier des manifestations qui entretiennent un rapport avec le transfert de technologie. Il sera indiqué la date de début , la période, le titre, un résumé explicatif sur la manifestation et les contacts concernant cette manifestation.

Enfin le service proposera une messagerie à certains utilisateurs privilègiés par le biais d'un choix caché.

ORGANISATION DES DONNEES DANS LES FICHIERS INFORMATIQUES.

La CRCI et l'ARIST ont de très nombreux fichiers informatisés. Ainsi l'ensemble des informations contenues par une fiche du service ARISTOTELE sont dispersés dans pas moins de 9 fichiers informatiques différents ce qui n'est pas sans poser parfois quelques problèmes pour rassembler l'ensemble de ces informations dans un même fichier afin de pouvoir charger ou mettre à jour la base de données. Nous verrons ceci plus loin.

Cependant afin de donner une idée de l'organisation des données nous foournissons à nouveau la fiche de l'ARIST avec en parralèle le nom des fichiers où sont puisées les informations.

SIGLE : ARIST RA (Fichier ARI.COMM)

NOM DE L'ORGANISME :Agence Régionale d'INformation Scientifique et Technique Rhône Alpes (Fichier CR.OBSER)

ADRESSE : (Fichier CR.IDREL) Quai Achille Lignon

LYON

69459 LYON CEDEX 06

Téléphone: 78 89 29 29

Telex : 900677 CRCIROL

CONTACTS : (Fichier CR.IDEIN) Mr Vuaillat Michel

Directeur

(Fichier ARI.TIT)Poste 140

(Fichier CR.CODIF)

Mme DESCHARMES Sylvianne Ingénieur

Poste 142

DOMAINE : (Fichier AR.SISTA)

Pluridisciplinaire

TYPES : (Fichier CR.IDREL)

Relais et appuis des entreprises;Chambre de Commerce et d'Industrie.

MOTS CLES : (Fichier AR.SISMC)

IST ; BASE DE DONNEES ; PROPRIETE INDUSTRIELLE ; TRANSFERT TECHNOLOGIE ;

COMMENTAIRE : (Fichier ARI.COMM)

Pour toute entreprise, la bonne information, livrée au bon moment, donne aujourd'hui la clé de l'efficacité. Les ARIST ont pour vocation de répondre de la façon la plus performante à la demande en information scientifique et économique des entreprises.

2) Mon travail sur ARISTOTELE

La première chose à faire était de vérifier systèmatiquement la validité des informations que nous serions appelés à utiliser dans les différents fichiers. Dans un premier temps je suis parti du fichier principal d'organismes que l'ARIST a constitué au cours de son travail le fichier SISTRA. De ce fichier informatique j'ai extrait tous les organismes que l'ARIST avait recensés en Rhône Alpes puisque le service ARISTOTELE est centré sur la région Rhône Alpes. Avec la liste ainsi constituée j'ai extrait dans deux grands fichiers la fiche de chaque organisme (les deux fichiers récapitulatifs sont SISTRA de l'ARIST et GENERA de la CRCI) et j'ai imprimé donc deux fiches de renseignements pour chaque organisme. Ensuite

j'ai vérifié systèmatiquement la validité des informations contenues dans les fichiers informatiques en les confrontant avec le dossier papier que possède l'ARIST sur chacun de ces organismes . Ces dossiers papiers sont régulièrement mis à jour grace à l'information reçue quotidiennement et grace aussi aux réponses aux demandes d'information régulières que fait l'ARIST auprès de ces organismes. Comme j'avais extrait une liste d'environ 550 organismes cela représentait un certain travail

mais cela m'a permis de me faire mieux connaître le tissu économique Rhône Alpes surtout en ce qui concerne le domaine des services. Toutes les rectifications à faire ont été notées sur les fiches papier que j'avais imprimé pour chaque organisme.

Pour chaque organisme il a aussi été nécessaire de lui attribuer d'une part un code catégorie et d'autre part un code domaine ce qui permettrait ensuite au logiciel de gestion de rechercher les fiches par catégorie ou par domaine.

Voir ci après la liste des domaines appelés champ secteur et la liste des catégories appelées champ type. Nous n'avons pas attribué de champ type aux organismes qui n'ont pas été retenus pour être dans ARISTOTELE. Il est à noter que je n'ai pas participé à l'élaboration de ces listes, les choix avaient été faits avant que j'arrive en stage au moment de la redéfinition du service.

TABLES DE CORRESPONDANCE ENTRE CODES ET LIBELLES

es champs TYPE et SECTEUR seront transmis sous la forme d'un code.
tant donné que l'affichage à l'écran de la fiche doit se faire avec
n libellé en clair, voici ci-dessous les tables de correspondance.

hamp SECTEUR

haque fiche ne contient qu'un seul secteur. Le champ SECTEUR sur
otre fichier de transfert contiendra un code à 4 caractères

l'affichage des fiches, ce code devra être remplacé par les
ibellés suivants.

GRO = Agriculture Agroalimentaire Agronomie
JTO = Automat. Robot. Informatique industr.
:P = Batiment Travaux-publics Urbanisme
IIM = Chimie Parach. Traitements de surf.
.EC = Electric. Electro. Physique de base
:ST = Gestion Normalisation Qualité
BA = Emballage Conditionnement
'ER = Energie Environnement Pollution
'FO = Informat. de Gestion Télecom. Télémat
.TE = Matériaux Métallurgie Sidérurgie
CA = Mécanique Travail des Matériaux
SU = Mesure Test Contrôle
NT = Santé Médecine Pharmacie Sécurité
XT = Textile Habillement
AN = Transports Manutention Stockage
UR = Pluridisciplinaire

AMP TYPE

aque fiche pourra contenir jusqu'à 3 types de 2 ou 4 caractères.

l'affichage des fiches, les codes devront être remplacés par les
bellés suivants :

= Pôles de compétences
AS = Pôles de compétences
Associations
CT = Pôles de compétences
Centres techniques
CR = Pôles de compétences
CRITT
PO = Pôles de compétences

Pôles technolPôles de compétences
ogiques
ATGR = Pôles de compétences
Groupements scientifiques
ATRE = Pôles de compétences
Recherche publique
ATCC = Pôles de compétences
Centres communs ou internationaux

EN = Enseignement supérieur
ENUN = Enseignement supérieur
Universités
ENGE = Enseignement supérieur
Grandes Ecoles
ENIU = Enseignement supérieur
IUT
ENLY = Enseignement supérieur
Lycées techniques
ENJU = Enseignement supérieur
Junior entreprises des Grandes Ecoles ou d'Universités
ENIN = Enseignement supérieur
Interfaces Enseignement supérieur - Industrie

IN = Information et documentation
INCD = Information et documentation
Centres de documentation et Bibliothèques
INPR = Information et documentation
Presse régionale
INBD = Information et documentation
Producteurs de bases de données

FI = Financement
FIIN = Financement
Aides financières publiques à l'innovation
FISE = Financement
Aides financières publiques de recours aux conseils
FICR = Financement
Sociétés financières d'innovation et de capital risque
FIBA = Financement
Banques et organismes de crédit

IP = Institutions et organismes publics
IPIR = Institutions et organismes publics
Institutions publiques régionales
IPID = Institutions et organismes publics
Institutions publiques départementales
IPIL = Institutions et organismes publics
Institutions publiques locales
IPOR = Institutions et organismes publics
Organismes de développement régionaux
IPOD = Institutions et organismes publics

Organismes de développement départementaux
:POL = Institutions et organismes publics
Organismes de développement locaux

:E = Relais et appuis des entreprises
:ECC = Relais et appuis des entreprises
Service Innovation des Chambres de Commerce et d'industrie
:EIN = Relais et appuis des entreprises
CRITT généralistes.
EFE = Relais et appuis des entreprises
Fédérations et groupements professionnels
EPA = Relais et appuis des entreprises
Organismes patronaux
EAU = Relais et appuis des entreprises
Autres structures d'appui

Au terme de cette première partie du travail je me suis retrouvé à la tête d'environ 350 fiches vérifiées et classées par catégories et domaines. Il m'a fallu alors rentrer les modifications dans les fichiers informatiques ce qui a représenté aussi un certain travail de saisie.

De plus au cours de la vérification j'ai été amené à faire de l'indexation car j'ai profité de la saisie pour enrichir les fiches de nouveaux mots clés lorsque c'était nécessaire. A ce propos on peut noter que la liste des mots clés, on ne peut pas parler de thésaurus puisqu'il n'y a pas de hiérarchie entre les mots clés, est bien faite et sert à la fois à indexer les organismes fichés par l'ARIST et les articles ou ouvrages qui se trouvent dans la base de données interne de l'ARIST. Cette base de donnée nommée ARISTODOC contient déjà 7000 références et s'enrichit tous les jours elle est très utile pour les réponses rapides aux questions urgentes et limitées ou pour les ingénieurs afin de démarrer leurs dossiers. Nous n'en parlerons pas plus parce qu'elle entretenait peu de rapports avec mon stage et ce serait alourdir ce mémoire.

Enfin le fichier ARI.COMM destiné à recueillir les sigles et les commentaires des organismes n'existait pas il m'a donc fallu le constituer. Pour chaque organisme j'ai crèé une fiche et lorsque c'était nécessaire et possible j'ai rédigé une petite fiche de commentaire synthètique pour l'organisme en question. J'ai doté environ 150 organismes de leur commentaire ce qui représente un certain travail de rédaction. Je ne l'ai pas fait pour tout le fichier car c'est un travail qui aurait dépassé

largement mes forces et le temps qui m'était imparti, d'autres part nous avons réalisé un mailing pour inviter les organismes cités dans le service à consulter leur fiche et à rédiger éventuellement le commentaire qui leur convient. Il fallait en faire un certain nombre et de façon assez variée pour que chacun puisse voir comment tirer profit de cette zone commentaire.

Une fois cette première base du fichier définie et assurée il fallait encore enrichir le service avec des organismes qui nous auraient échappés parce qu'ils ne se trouvaient pas dans le premier fichier de l'ARIST. J'ai ainsi été amené à passer en revue l'ensemble des annuaires que possède l'ARIST pour repèrer les organismes qui pourraient nous manquer , de même j'ai interrogé différentes personnes de la CRCI spécialistes de tel ou tel secteur pour voir les organimes oubliés. Le fichier s'est ainsi enrichi d'une centaine de fiches pour lesquelles il a fallu aussi saisir tous les renseignements. Le fichier constitué représente environ 450 fiches d'organismes.

Au cours du mois de Juillet nous avons fait un premier chargement du service avec un jeu d'une cinquantaine de fiches de façon à essayer pleinement le service et en relever les défauts , les limites, les erreurs.

J'ai donc essayé systèmatiquement le service sur minitel pour en repèrer toutes les insuffisances. De même toutes les personnes de l'ARIST ont été invitées à effectuer des essais

similaires et ensuite j'ai été amené à synthètiser l'ensemble des problèmes relevés afin d'en faire part au serveur. Voir le document ci-après que j'ai rédigé à l'attention du serveur.

ESSAIS D'ARISTOTELE : CRITIQUES.

GE D'INTRODUCTION

- Anomalies de fonctionnement:

1 - Le numero de l'option choisie ne s'affiche pas à l'écran et
me le démarrage est assez lent à ce niveau l'utilisateur se
nande toujours pendant un moment si le service a bien intègré sa
nande.

Modifications d'affichage souhaitées:

1 - LOGO: Il y a un carré sombre sous la jambe du R dans la
:tie claire ,il faudrait le supprimer.

2 - TEXTE DES OPTIONS: Il faudrait remplacer 'Interroger
'STOTELE' par 'Interrogation directe d'ARISTOTELE ' et remplacer à
ption 2 'Consulter ARISTOTELE' par 'Interrogation guidée
RISTOTELE'.

'ION 1:

Anomalies de fonctionnement:

1 - Pour les rubriques Secteur ou Type , on peut taper
ectement le code fourni par l'ARIST et connu d'elle seule mais
t on taper le numéro correspondant à la page de guide. Dans les
x cas, peut on faire afficher le libellé long dans la grille à la
ce du code ou du numéro.

2 - Dans la rubrique type d'organisme lorsqu'on est dans le
de on ne peut pas voir immédiatement l'ensemble des réponses qui
t proposées au premier niveau , on est obligé d'affiner la
ection jusqu'au deuxième niveau.

3 - Dans la rubrique mots clefs lorsqu'on demande d'affiner la
herche le service efface les mots clefs déjà inscrits.

4 - Lorsqu'on est dans les pages de guide il indique parfois
er votre numero et envoi alors qu'il n'y a manifestement pas de
éro à choisir (rubrique Nom ou Sigle).

5 - Enfin lorsqu'on selectionne un secteur ou un type d'activité il y a 0 réponse le service n'affiche pas sur la grille en clair secteur ou le type demandé.

6 - Lorsqu'on est dans l'affichage long ou court si on veut .fier la technique de recherche on est en fait obligé de la ommencer complètement depuis le début.Ne serait il pas possible de nir par des retours successifs aux étapes de définition de erche intemédiaire

7 - Lorsqu'on affine une question en repassant par retour sur ligne déjà remplie, cette ligne est effacée même si on désire la erver.

8 - Si on met par exemple Département 69 et mots clefs Plastique ue l'on fait retour pour rajouter d'autres critères le service che directement la fiche en format max (il n'y en a qu'un) au d'afficher comme d'habitude vous avez x réponses.

9 - Si on rentre un département et un type et que l'on fait ur, au lieu de venir sur secteur d'activité on aboutit sur les clefs . En règle générale si on rentre un critère et que l'on retour, c'est suite qui se fait.

10 - La troncature ne se fait qu'à partir de trois caractères le nom (normal je pense).

Modification d'affichage souhaitées.

1 - Lorsqu'on est dans l'affichage court ou long il manque nt (elle est là quelquefois) l'affichage de la touche suite avec ôle alors que la fonction est essentielle pour voir l'ensemble information.

2 - Lorsqu'on est dans l'affichage long si on est à la page 1 on tour) pour revenir aux documents courts et si l'on est a la 2 ou 3 on a la commande (* retour): serait il possible d'unifier ommandes , d'en avoir une seule toujours la même.

3 - Sur la grille de saisie il faudrait supprimer les mots on ou',conserver 'Département'.

4 - Dans l'affichage court ou long il faudrait remplacer les s 'affichage court' par 'Liste des organismes' et 'Affichage par 'Fiche complète'.

5 - Dans affichage long dans les touches d'aide au bas de an il faudrait remplacer liste des documents par liste des ismes.

6 - Dans les pages de guide certains libellés OSCAT doivent être ies (Liste en annexe).

7 - Dans les pages de guide certaines lignes pourront être primées (liste à définir).

8 - Dans la page guide des secteurs serait il possible de mettre vidéo inverse ou dans une autre couleur le premier mot de chaque ne afin de le faire ressortir.

ION 2:

Anomalies de fonctionnement:

1 - Grosse anomalie: dans la grille des types d'organisme ɔtion 4 et l'option 5 Institutions et Organismes publics renvoient la même sous grille celle des Institutions publiques.

2 - Pour les rubriques Secteur ou Type , on peut taper ctement le code fourni par l'ARIST et connu d'elle seule mais on taper le numéro correspondant à la page de guide. Dans les cas, peut on faire afficher le libellé long dans la grille à la e du code ou du numéro.(Idem que dans l'OPTION 1)

3 - Dans la rubrique type d'organisme lorsqu'on est dans le e on ne peut pas voir immédiatement l'ensemble des réponses qui proposées au premier niveau , on est obligé d'affiner la ction jusqu'au deuxième niveau.(Idem que dans l'OPTION 1)

4 - Lorsqu'on est dans l'affichage long ou court si on veut fier la technique de recherche on est en fait obligé de la mmencer complètement depuis le début.Ne serait il pas possible de nir par des retours successifs aux étapes de définition de erche intermédiaires(Idem que dans l'OPTION 1)

Modification d'affichage souhaitées.

1 - Lorsqu'on est dans l'affichage court ou long il manque ent (elle est là quelquefois) l'affichage de la touche suite avec rôle alors que la fonction est essentielle pour voir l'ensemble 'information.(Idem que dans l'OPTION 1)

2 - Lorsqu'on est dans l'affichage long si on est à la page 1 on etour) pour revenir aux documents courts et si l'on est a la 2 ou 3 on a la commande (* retour): serait il possible d'unifier commandes , d'en avoir une seule toujours la même.(Idem que dans TION 1)

3 - Dans l'affichage court ou long il faudrait remplacer les es 'affichage court' par 'liste des organismes' et 'Affichage ' par 'Fiche complète'.

4 - Le logo n'apparait pas en dynamique dans cette partie du service

5 - Est il possible de remettre 'Interrogation guidée d'ARISTOTELE' sur l'ensemble des pages de cette OPTION.

6 - Dans affichage long dans les touches d'aide au bas de l'écran il faudrait remplacer 'liste des documents' par 'liste des organismes'.

7 - Dans les pages de guide certains libellés SSCAT doivent être modifiés (Liste en annexe).

8 - Dans les pages de guide certaines lignes pourront être supprimées (liste à définir).

9 - Dans la page guide des secteurs serait il possible de mettre en vidéo inverse ou dans une autre couleur le premier mot de chaque ligne, afin de le faire ressortir.

)PTION 3:

1 - Pas d'anomalies de fonctionnement constatées.

2 - Cependant il faudrait modifier l'affichage pour que ce soit)lus clair : dans un premier temps afficher 'Ecrivez votre nom et 'otre adresse' et ensuite au lieu de 'envoi de message' afficher Texte de votre message'.

)PTION 4:

1 - Les manifestations sont supprimées trop tôt , en effet le endemain du début.Pourrait on les effacer comme convenu environ une emaine après la première date pour leur laisser le temps de se erminer.

'PTION 5:

Pourrait on supprimer l'option BAL à partir de cette option.

'PTION SECRET:

1 - Messagerie : pas de problèmes.

2 - Saisie des manifestations: Pas problèmes:

Enfin je me suis occupé de repèrer les manifestations ou les conférences ou les colloques entretenant des rapports avec le transfert de technologie et j'ai fait un premier chargement de cette partie du service. En effet il est possible de saisir directement les manifestations à partir de l'option secrète offerte à certains utilisateurs privilègiés.

III TRAVAIL DE PROGRAMMATION.

Au cours du stage j'ai été amené à programmer en Turbo Pascal par des circonstances particulières et j'ai considéré ceci comme une chance car cela m'a permis de varier mon travail et de mettre à profit les connaissances acquises en programmation et de les approfondir.

Il fallait rassembler les informations dispersées dans les différents fichiers dans un seul et même fichier afin de les fournir au serveur le SUNIST. Ce travail a été réalisé par monsieur Puget l'informaticien et le documentaliste de l'ARIST. Il a fait ce travail avec QRY un logiciel d'analyse de fichiers sur mini ordinateur IBM, ce qui m'a permis en outre de commencer à me familiariser avec ce logiciel. Cependant QRY fournit des fichiers textes qui devaient être modifiés pour être acceptés par le SUNIST. En effet le fichier fourni par QRY comprenait des lignes de 1300 caractères séparées par des retours chariots où les zones sont collées bout à bout sans séparateur ni indication des zones. Le contrat signé avec le SUNIST prévoyait que chaque zone de chaque enregistrement serait séparée par un retour chariot et d'autre part chaque zone devait être précèdée de son nom. Il fallait donc faire subir au fichier un traitement informatique avant de pouvoir le charger. Le SUNIST était prêt à développer le programme de reformatage nécessaire mais contre

rémunèration et réclamait un premier délai d'une semaine ce qui par divers décalages aurait repoussé les essais après les congés d'été , le service aurait pris un certain retard. Comme je disposais du langage Turbo Pascal j'ai essayé d'écrire un petit programme permettant de faire ce reformatage. Nous avons transfèré le fichier de l'IBM 36 sur le disque dur d'un PC XT grace au logiciel de transfert GPC.

Mon premier programme comme vous pourrez le voir dans les annexes est assez fruste et inélégant mais il a eu le mérite d'être écrit en une demie journée et de fonctionner ce qui a levé immédiatement l'hypothèque d'un retard.

Dans un deuxième temps je me suis attaché à rendre ce programme plus souple de façon à règler l'ensemble des zones grace à un algorithme plus court et plus élégant. J'ai du écrire aussi une procèdure capable d'examiner les mots afin de placer les retours chariots au bon endroit car ma première version avait tendance à couper les mots en plein milieu, ceci avait entre autre le résultat facheux que le logiciel du sunist multipliait les mots clés avec les demis mots parfois fournis. J'ai toujours procèdé par la mise au point de versions successives ce qui permet de voir le fonctionnement et l'avancée du travail à chaque fois.(voir annexes)

Dans un deuxième temps Monsieur Puget m'a demandé d'améliorer le programme dans le sens d'une plus grande souplesse d'utilisation. En effet ce qui pouvait être interessant c'était de permettre à l'utilisateur de définir lui

même le nom des zones, leur longueur, s'il était nécessaire d'afficher la zone dans le fichier. Encore mieux il fallait permettre à l'utilisateur de conserver plusieurs options de formatage et de pouvoir les utiliser selon les besoins. Dans un premier temps il me fallait modifier mon programme pour que la procèdure de formatage aille chercher les paramètres de chaque zone dans un tableau, puis ensuite d'écrire un module de définition de formatage qui sauvegarde les paramètres dans un fichier. De fait il fallait que monsieur Puget puisse continuer à se servir du logiciel après que je sois parti même si les conditions du formatage étaient appelées à changer.

Il fallait aussi rendre le logiciel plus ergonomique.

Le module de définition de format permet de choisir le nombre de zones, de choisir leur noms, de choisir de les afficher ou pas, de définir s'il est nécessaire ou pas de chercher les blancs pour faire les coupures en fin de ligne.

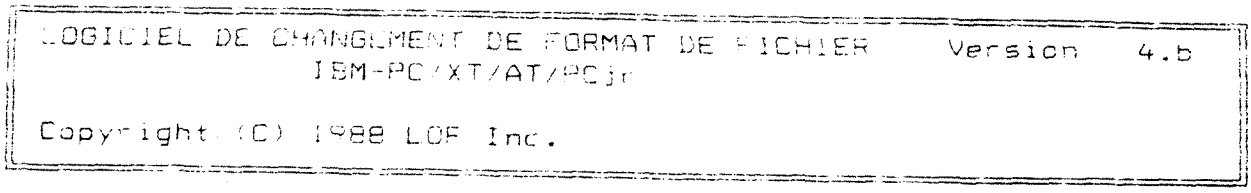
Afin d'améliorer l'ergonomie du programme j'ai mis au point une procèdure qui nomme le nouveau fichier formatté avec le nom du fichier de départ et en ajoutant l'extension .ARR; De même le module qui permet la définition d'un fichier de paramètres ajoute automatiquement l'extension .ANA au nom choisi (.ANA pour analyse) ce qui permet à l'utilisateur de s'y retrouver plus facilement dans ses fichiers qui ont comme tout ce qui est vivant tendance à se multiplier.

Je voulais rendre encore le programme beaucoup plus facile d'accès et d'utilisation ce qui m'a amené à creuser la question de l'interface utilisateur. Comment j'ai règlé le problème. j'étais incapable d'écrire de but en blanc une interface

utilisateur qui ait une véritable allure professionnelle. C'est un travail long et compliqué. Aussi j'ai fait travail de recherche documentaire dans les routines écrites en Pascal. J'ai trouvé ainsi un programme fourni par Borland avec son code source. Ce programme est fourni dans le module Turbo Tutor destiné à montrer toutes les possibilités de Turbo Pascal. J'ai donc trouvé le logiciel Filemgr.pas qui est un logiciel de gestion de fichiers et de répertoires. Ce logiciel avait une interface très conviviale je l'ai donc repris à mon compte. J'ai enlevé les fonctions qui ne m'interessaient pas (par exemple effacer un fichier ou détruire un répertoire), j'ai gardé les fonctions qui m'interessaient (par exemple la possiblité de choisir un fichier dans une liste en faisant bouger un bandeau en vidéo inverse sur les noms et enfin j'ai intègré mes propres procèdures. Ceci m'a permis d'écrire un programme très convivial avec une belle allure à peu de frais. Ceci m'a fait beaucoup progresser car adapter des procèdures demande toujours finalement de se plonger dedans pour les comprendre car il devient toujours nécessaire de les modifier légèrement pour les adapter à ses besoins. J'ai été amené à découvrir ce qu'étaient la gestion des interruptions du processeur 8088 et du coup j'ai été amené à essayer de comprendre le langage machine du PC. De fil en aiguille j'ai été obligé de lire un certain nombre d'ouvrages sur le pascal qui à leur tour m'ont apporté d'autres routines et manières de faire. Ceci me permettra de programmer plus vite et plus efficace. Aujourd'hui d'ailleurs je peux voir certaines limites de mon programme, j'aurais du m'abstenir d'utiliser la commande window qui n'est pas portable sur tous

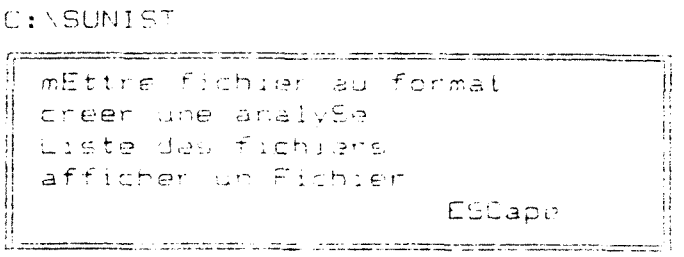
les micros, elle empèche notamment de travailler sur les micros dotés d'une carte Hercule. J'aurais pu obtenir les mêmes résultats d'affichage en utilisant les techniques de sauvegarde d'écran et en même temps j'aurais obtenu un programme plus portable. Cette faiblesse de mon programme en l'occurence n'est pas très importante car il est destiné à ne servir que dans le cadre de la CRCI dont tous les micros disposent d'une carte CGA ou EGA.

Première page offerte par le logiciel au démarrage. La fenètre s'efface automatiquement au bout de quelques secondes ou si l'utilisateur enfonce une touche.

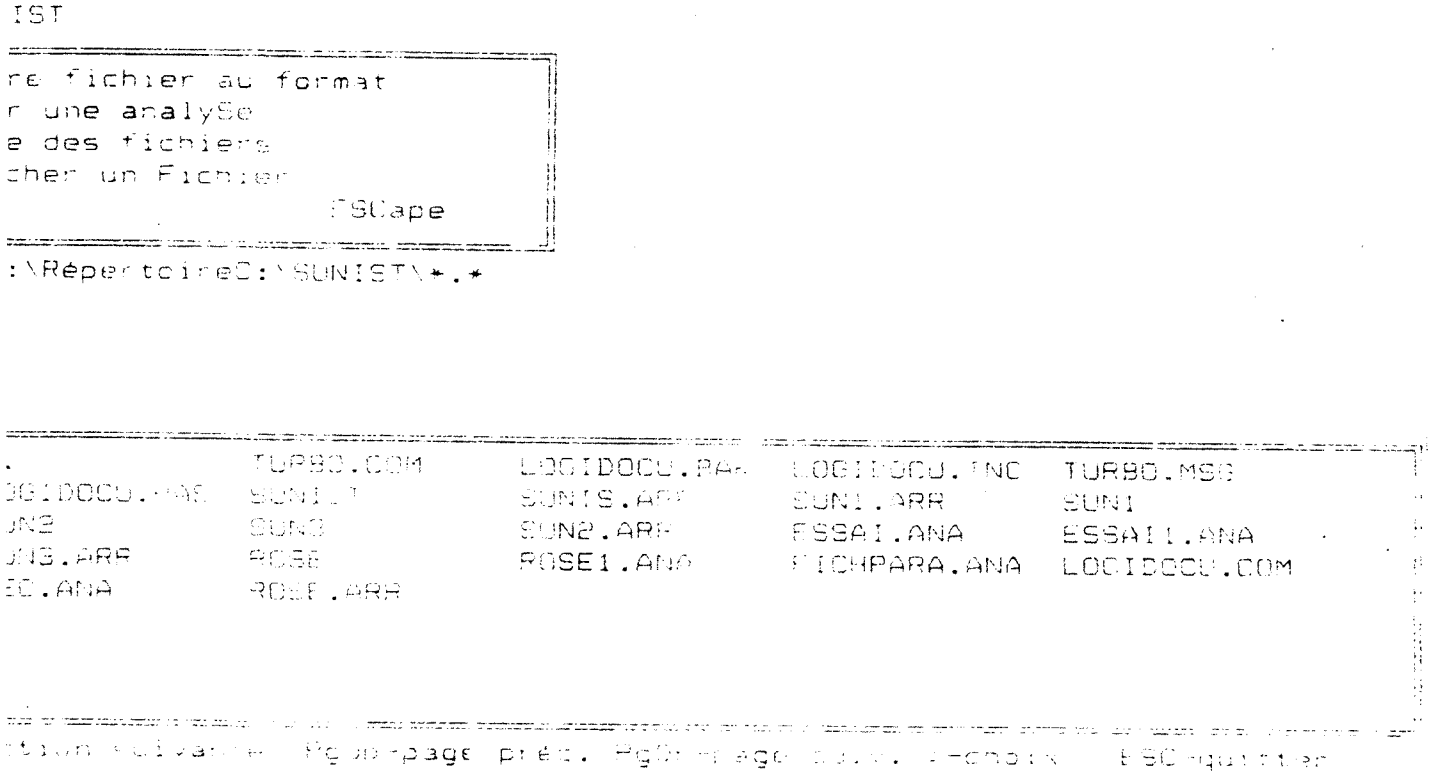


Fenêtre autorisant le choix d'une action ou d'une autre. On voit facilement qu'il suffit de taper la lettre mise en majuscule pour lancer l'opération .

Au dessus de la fenêtre est affiché automatiquement le répertoire courant.



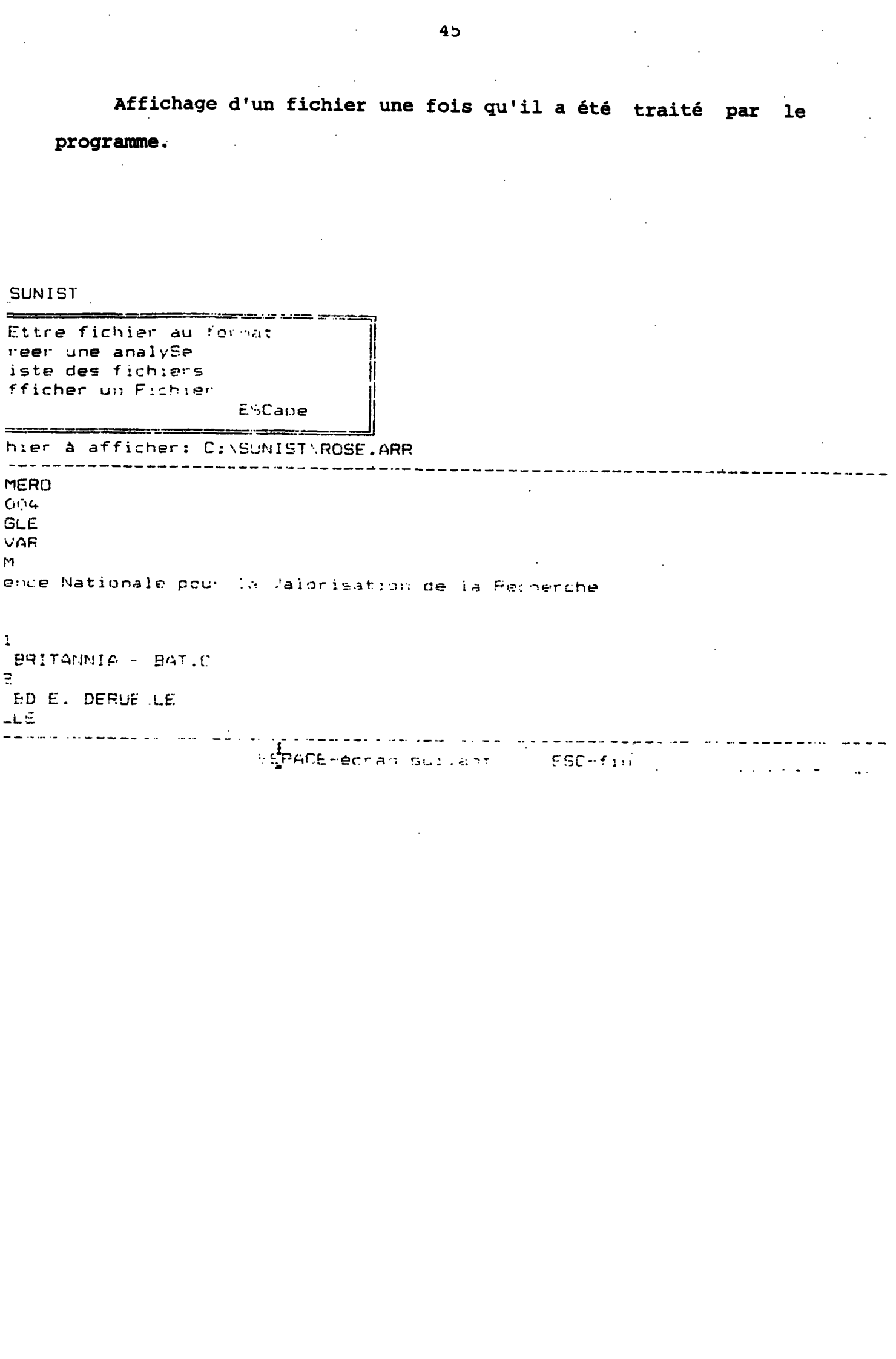
Ce qu'on voit lorsqu'on demande la liste des fichiers. Si on tape entrée lorsque le bandeau inverse est sur les deux point on remonte à la racine . Si on tape entrée lorsque le bandeau est sur un nom de répertoire le programme affiche les fichiers du nouveau répertoire. Une ligne d'aide est affichée en bas de l'écran.



Affichage du fichier SUNIST tel qu'il est fourni par le PS36, on ne voit ici que les 78 premiers caratères mais on peut voir comment il est organisé et comment tous les renseignements sur un même organisme sont sur une même ligne.

```
IST
re fichier au format
r une analyse
e des fichiers
cher un fichier
                    ESCape
r à afficher: C:\SUNIST\SUNIST
-------------------------------------------------------------------------
                    Région Rhône-Alpes
ADEMAP              Agence pour le Developpement de la Mise en oeuvre des
ADERLY              Association pour le Développement Economique de la Rè
ADERME              Association pour le Développement de l'Enseignement e
ADIL                Association pour le Développement Industriel de la Lo
ANVAR               Agence Nationale pour la Valorisation de la Recherche
CALFETMAT           Centre d'Applications des Lasers de Forte Energie à L
CCI Valence         Chambre de Commerce et d'Industrie de Valence
CCIL                Chambre de Commerce et d'Industrie de Lyon
CETIM               Centre Technique des Industries Mécaniques
CIRC                Centre International de Recherche sur le Cancer
CTDEC               Centre Technique de l'Industrie du Décolletage
FSLSE               Fondation Scientifique de Lyon et du Sud Est
-------------------------------------------------------------------------
                ESPACE-écran suivant      ESC-fin
```

Affichage d'un fichier une fois qu'il a été traité par le programme.



En conclusion je voudrais remercier tous les gens de l'ARIST avec qui j'ai travaillé pour la gentillesse avec laquelle ils m'ont accueilli. Je voudrais remercier tout particulièrement Monsieur Puget avec qui j'ai travaillé tous les jours sur le projet ARISTOTELE. outre sa sympathie et son amabilité j'ai pu apprècié son sèrieux et son professionnalisme qui m'ont beaucoup apporté sur le plan professionnel et humain.

Dans cette annexe nous trouverons les différentes versions des programmes que j'ai écrits en Turbo Pascal. Le véritable programme définitif commence à la page 21 Sunlog.inc.

```
program sunist;  (* Premiére version du programme *)

var D,A:text; I,Compt:integer; V:array [1..500] of char;Ch:char;


Procedure Decoupe;        (* Découpe une zone dans le fichier de
départ et l'écrit dans le fichier d'arrivée *)
 begin
                  for I:=1 to Compt do
                      begin
                      read(D,Ch);
                      V[I]:=(Ch);
                      end;
                  for I:=1 to Compt do
                      begin
                      write(A,V[I]);
                      end; end;


begin
     for I:=1 to 500 do V[I]:=' ';
     assign(D,'sunist');              (* Dans cette version le nom
du fichier est inscrit dans le programme , gestion peu souple *)
     reset(D);
     assign(A,'arrivee');
     rewrite(A);
           while not EOF(D) do
                  begin
                  write(A,'NUMERO');   (* Les noms de zones sont
écrits dans le programme , pas souple non plus *)
                  writeln(A,#13);
                  compt:=5;
                  Decoupe;
                  writeln(A,#13);
                  write(A,'SIGLE');
                  writeln(A,#13);
                  compt:=20;
                  Decoupe;
                  writeln(A,#13);
                  write(A,'NOM');
                  writeln(A,#13);
                  compt:=80;
                  Decoupe;writeln(A,#13);
                  Decoupe;writeln(A,#13);
                  compt:=26;
                  Decoupe;writeln(A,#13);
                  write(A,'AD1');
                  writeln(A,#13);
                  compt:=40;
                  Decoupe;
```

```
writeln(A,#13);
write(A,'AD2');
writeln(A,#13);
compt:=40;
Decoupe;
writeln(A,#13);
write(A,'VILLE');
writeln(A,#13);
compt:=30;
Decoupe;
writeln(A,#13);
write(A,'CODBUDIS');
writeln(A,#13);
compt:=40;
Decoupe;
writeln(A,#13);
write(A,'TELEP');
writeln(A,#13);
compt:=20;
Decoupe;
writeln(A,#13);
write(A,'TELEX');
writeln(A,#13);
compt:=15;
Decoupe;
writeln(A,#13);
write(A,'CONTACT1');
writeln(A,#13);
compt:=40;
Decoupe;
writeln(A,#13);
write(A,'FONCT1');
writeln(A,#13);
compt:=30;
Decoupe;
writeln(A,#13);
write(A,'TELEP1');
writeln(A,#13);
compt:=15;
Decoupe;
writeln(A,#13);
write(A,'CONTACT2');
writeln(A,#13);
compt:=40;
Decoupe;
writeln(A,#13);
write(A,'FONCT2');
writeln(A,#13);
compt:=30;
Decoupe;
writeln(A,#13);
write(A,'TELEP2');
writeln(A,#13);
compt:=15;
Decoupe;
writeln(A,#13);
write(A,'SECTEUR');
```

```
writeln(A,#13);
compt:=4;
Decoupe;
writeln(A,#13);
write(A,'TYPE');
writeln(A,#13);
compt:=18;
Decoupe;
writeln(A,#13);
write(A,'MOTSCLES');
writeln(A,#13);
compt:=80;
Decoupe;writeln(A,#13);
Decoupe;writeln(A,#13);
Decoupe;writeln(A,#13);
Decoupe;writeln(A,#13);
Decoupe;writeln(A,#13);
compt:=53;
Decoupe;writeln(A,#13);
write(A,'COMMENT');
writeln(A,#13);
compt:=80;
Decoupe;writeln(A,#13);
Decoupe;writeln(A,#13);
Decoupe;writeln(A,#13);
compt:=16;
Decoupe;writeln(A,#13);
Compt:=2;
Decoupe;
writeln(A,#13);
readln(D);
end;
close(D);
close(A);
delay(500);
reset(A);
while not EOF(A) do
begin
read(A,Ch);
write (Ch);
end;
end.
```

```
program sunist1;

var D,A:text; I,Compt,J:integer; V:array [1..500] of char;Ch:char;
TCompt:array[1..20] of integer;TNom:array[1..19]of string[30];

Procedure Decoupe;
 begin
                  for I:=1 to (TCompt[J]) do
                      begin
                      read(D,Ch);
                      V[I]:=(Ch);
                      end;
                  for I:=1 to (TCompt[J]) do
                      begin
                      write(A,V[I]);
                      end; end;



Procedure Decoupe1;
 begin
                  for I:=1 to Compt do
                      begin
                      read(D,Ch);
                      V[I]:=(Ch);
                      end;
                  for I:=1 to Compt do
                      begin
                      write(A,V[I]);
                      end; end;



Procedure RempliTCompt;       (* Dans cette version on utilise des
tableaux pour conserver les paramètres, c'est plus joli que dans
le corps du programme *)

begin
  TCompt[1]:=5;TCompt[2]:=20; TCompt[3]:=186;TCompt[4]:=40;
  TCompt[5]:=40;TCompt[6]:=30; TCompt[7]:=40;TCompt[8]:=20;
  TCompt[9]:=15;TCompt[10]:=40; TCompt[11]:=30;TCompt[12]:=15;
  TCompt[13]:=40;TCompt[14]:=30; TCompt[15]:=15;TCompt[16]:=4;
  TCompt[17]:=18;TCompt[18]:=453; TCompt[19]:=256;TCompt[20]:=2;
end;


Procedure RempliTNom;

begin
  TNom[1]:='NUMERO';TNom[2]:='SIGLE';
TNom[3]:='NOM';TNom[4]:='AD1';
```

```
  TNom[5]:='AD2';TNom[6]:='VILLE';
TNom[7]:='CODBUDIS';TNom[8]:='TELEP';
  TNom[9]:='TELEX';TNom[10]:='CONTACT1';
TNom[11]:='FONCT1';TNom[12]:='TELEP1';
  TNom[13]:='CONTACT2';TNom[14]:='FONCT2';
TNom[15]:='TELEP2';TNom[16]:='SECTEUR';
  TNom[17]:='TYPE';TNom[18]:='MOTSCLES'; TNom[19]:='COMMENT'; end;

Procedure DecoupeZone;

 begin
                  for J:=1 to 19 do
                  begin
                  write(A,(TNom[J]));
                  writeln(A,#13);
                  Decoupe;
                  writeln(A,#13);
                  end;
 end;



begin
     for I:=1 to 500 do V[I]:=' ';
     assign(D,'sunist');
     reset(D);
     assign(A,'arrive1');
     rewrite(A);
     RempliTCompt;
     RempliTNom;
           while not EOF(D) do
                  begin
                  DecoupeZone;
                  Compt:=2;
                  Decoupe1;
                  writeln(A,#13);
                  readln(D);
                  end;


close(D); close(A); write ('Le Programme est terminé lisez le
fichier arrive1'); delay(500); reset(A); while not EOF(A) do begin
read(A,Ch); write (Ch); delay(10); end;

end.
```

```
program sunist2;          (* Dans cette version on n'a pas
installé à nouveau la boucle de la version précèdente, on la
reprendra dans les autres versions *)

var D,A:text; I,Compt,Dep,ComptZ,K,Increment,J:integer; V:array
[1..500] of char;Ch:char; Comptage :array [1..10] of integer;


Procedure CR; begin
     writeln(A,#13); end;


Procedure Lire;
 begin
          for I:=1 to Compt do
              begin
              read(D,Ch);
              V[I]:=(Ch);
              end;
 end;

Procedure Ecrire(var Dep:integer);      (* on a divisé la procédure
découpe en deux procédures *)

begin
             for I:=Dep to Compt do
                 begin
                 write(A,V[I]);
                 end;
                 CR; end;


Procedure Decoupe;

begin
     Lire;
     Dep:=1;
     Ecrire(Dep); end;


Procedure ChercheBlanc;       (* Permet de repèrer où mettre les
retours chariots *)

begin
     repeat
     ComptZ:=ComptZ-1;
     until V[ComptZ+1]=#32; end;

Procedure RemplitComptageZero;  (* Initialise un tableau qui
recevra les longueurs provisoires de zone *)

begin
     for K:=1 to 10 do Comptage[K]:=0; end;
```

```
Procedure RemplitComptage;

begin
      RemplitComptageZero;
      Increment:=81;ComptZ:=81;K:=1;
      while Increment>0 do
            begin
            If Increment=81 then CherchеBlanc;
            Comptage[K]:=ComptZ;
            Increment:=compt-ComptZ;
            if Increment>81 then Increment:=81;
            ComptZ:=ComptZ+Increment;
            K:=K+1;
            end; end;

Procedure Ecrire1;

begin
      RemplitComptage;
      K:=1;Dep:=1;
      while Comptage[K]>0 do
            begin
            compt:=Comptage[K];
            Ecrire(Dep);
            Dep:=Comptage[K]+1;
            K:=K+1;
            end; end;



begin clrscr; gotoxy(5,12);write('Patience je travaille ');
      for I:=1 to 500 do V[I]:=' ';
      assign(D,'sunist');
      reset(D);
      assign(A,'arrivee');
      rewrite(A);
            while not EOF(D) do
                  begin
                  write(A,'NUMERO');
                  CR;
                  compt:=5;
                  Decoupe;
                  write(A,'SIGLE');
                  CR;
                  compt:=20;
                  Decoupe;
                  write(A,'NOM');
                  CR;
                  compt:=186;
                  Lire;
                  Ecrire1;
                  write(A,'AD1');
                  CR;
                  compt:=40;
                  Decoupe;
                  write(A,'AD2');
```

```
CR;
compt:=40;
Decoupe;
write(A,'VILLE');
CR;
compt:=30;
Decoupe;
write(A,'CODBUDIS');
CR;
compt:=40;
Decoupe;
write(A,'TELEP');
CR;
compt:=20;
Decoupe;
write(A,'TELEX');
CR;
compt:=15;
Decoupe;
write(A,'CONTACT1');
CR;
compt:=40;
Decoupe;
write(A,'FONCT1');
CR;
compt:=30;
Decoupe;
write(A,'TELEP1');
CR;
compt:=15;
Decoupe;
write(A,'CONTACT2');
CR;
compt:=40;
Decoupe;
write(A,'FONCT2');
CR;
compt:=30;
Decoupe;
write(A,'TELEP2');
CR;
compt:=15;
Decoupe;
write(A,'SECTEUR');
CR;
compt:=4;
Decoupe;
write(A,'TYPE');
CR;
compt:=18;
Decoupe;
write(A,'MOTSCLES');
CR;
compt:=453;
Lire;
Ecrire1;
write(A,'COMMENT');
```

```
                    CR;
                    compt:=80;
                    Decoupe;writeln(A,#13);
                    Decoupe;writeln(A,#13);
                    Decoupe;writeln(A,#13);
                    compt:=16;
                    Decoupe;writeln(A,#13);
                    Compt:=2;
                    Decoupe;
                    writeln(A,#13);
                    readln(D);
                    end;
                    close(D);
                    close(A);
                    gotoxy(5,12);write('
');
                    gotoxy(5,12);writeln('C''est terminé, je vais
afficher le fichier formaté ');
                    delay(15000);
                    reset(A);
                    while not EOF(A) do
                    begin
                    read(A,Ch);
                    write (Ch);
                    end;
                    end.
```

```
program sunist3;

var D,A:text; I,Compt,J,Dep,ComptZ,K,Increment:integer; V:array
[1..500] of char;Ch:char; TCompt:array[1..20] of
integer;TNom:array[1..19]of string[30]; Comptage :array [1..10] of
integer;


Procedure CR; begin
     writeln(A,#13); end;


Procedure Lire;
 begin
          for I:=1 to Compt do
              begin
              read(D,Ch);
              V[I]:=(Ch);
              end;
 end;

Procedure Ecrire(var Dep:integer);

begin
             for I:=Dep to Compt do
                 begin
                 write(A,V[I]);
                 end;
                 CR; end;



Procedure Decoupe;

begin
     Lire;
     Dep:=1;
     Ecrire(Dep); end;


Procedure ChercheBlanc;

begin
     repeat
     ComptZ:=ComptZ-1;
     until V[ComptZ+1]=#32; end;

Procedure RemplitComptageZero;

begin
     for K:=1 to 10 do Comptage[K]:=0; end;

Procedure RemplitComptage;
```

```
begin
      RemplitComptageZero;
      Increment:=81;ComptZ:=81;K:=1;
      while Increment>0 do
            begin
            If Increment=81 then ChercheBlanc;
            Comptage[K]:=ComptZ;
            Increment:=compt-ComptZ;
            if Increment>81 then Increment:=81;
            ComptZ:=ComptZ+Increment;
            K:=K+1;
            end; end;

Procedure Ecrire1;

begin
      RemplitComptage;
      K:=1;Dep:=1;
      while Comptage[K]>0 do
            begin
            Compt:=Comptage[K];
            Ecrire(Dep);
            Dep:=Comptage[K]+1;
            K:=K+1;
            end; end;


Procedure RempliTCompt;

begin
  TCompt[1]:=5;TCompt[2]:=20; TCompt[3]:=186;TCompt[4]:=40;
  TCompt[5]:=40;TCompt[6]:=30; TCompt[7]:=40;TCompt[8]:=20;
  TCompt[9]:=15;TCompt[10]:=40; TCompt[11]:=30;TCompt[12]:=15;
  TCompt[13]:=40;TCompt[14]:=30; TCompt[15]:=15;TCompt[16]:=4;
  TCompt[17]:=18;TCompt[18]:=453; TCompt[19]:=256;TCompt[20]:=2;
end;


Procedure RempliTNom;

begin
  TNom[1]:='NUMERO';TNom[2]:='SIGLE';
TNom[3]:='NOM';TNom[4]:='AD1';
  TNom[5]:='AD2';TNom[6]:='VILLE';
TNom[7]:='CODBUDIS';TNom[8]:='TELEP';
  TNom[9]:='TELEX';TNom[10]:='CONTACT1';
TNom[11]:='FONCT1';TNom[12]:='TELEP1';
  TNom[13]:='CONTACT2';TNom[14]:='FONCT2';
TNom[15]:='TELEP2';TNom[16]:='SECTEUR';
  TNom[17]:='TYPE';TNom[18]:='MOTSCLES'; TNom[19]:='COMMENT'; end;




begin clrscr; gotoxy(5,12);write('Patience je travaille ');
```

```
      for I:=1 to 500 do V[I]:=' ';J:=1;
      assign(D,'sunist');
      reset(D);
      assign(A,'arrivee');
      rewrite(A);
      RempliTCompt;
      RempliTNom;
            while not EOF(D) do
                  begin
                        Dep:=1;
                        for J:=1 to 18 do
                        begin
                        write(A,TNom[J]);
                        CR;
                        Compt:=TCompt[J];
                        Lire;
                        if TCompt[J]>80 then
                        begin
                        Ecrire1;Dep:=1;
                        end
                        else Ecrire(Dep);
                        end;
                        write(A,'COMMENT');
                        CR;
                        Compt:=80;
                        Decoupe;
                        Decoupe;
                        Decoupe;
                        Compt:=16;
                        Decoupe;
                        Compt:=2;
                        Decoupe;
                        readln(D);
                  end;
                   close(D);
                   close(A);
                   gotoxy(5,12);write('
');
                   gotoxy(5,12);writeln('C''est terminé, je vais
afficher le fichier formaté ');
                   delay(5000);
                   reset(A);
                   while not EOF(A) do
                   begin
                   read(A,Ch);
                   write (Ch);
                   delay(10);
                   end;
 end.
```

```
program sunist4a;


type

Zone=record Nom:string[30]; AffNom:boolean; Long:integer;
AffLong:boolean; Q80:boolean; end;



var D,A:text; I,Taille,Compt,J,Dep,ComptZ,K,Increment,Num:integer;
V:array [1..500] of char;Ch:char; TCompt:array[1..50] of
integer;TNom:array[1..50] of string[30]; Comptage :array [1..10]
of integer; TAffNom:array[1..50] of boolean;TAffLong:array[1..50]
of boolean; TQ80:array[1..50] of boolean; Zone1:Zone; P:file of
Zone;

Procedure CR; begin
     writeln(A,#13); end;


Procedure Lire;
 begin
          for I:=1 to Compt do
              begin
              read(D,Ch);
              V[I]:=(Ch);
              end;
 end;

Procedure Ecrire(var Dep:integer);

begin
             for I:=Dep to Compt do
                 begin
                 write(A,V[I]);
                 end;
                 CR; end;



Procedure Decoupe;

begin
     Lire;
     Dep:=1;
     Ecrire(Dep); end;


Procedure ChercheBlanc;

begin
     repeat
     ComptZ:=ComptZ-1;
```

```
      until V[ComptZ+1]=#32; end;

Procedure RemplitComptageZero;

begin
      for K:=1 to 10 do Comptage[K]:=0; end;

Procedure RemplitComptage;

begin
      RemplitComptageZero;
      Increment:=81;ComptZ:=81;K:=1;
      while Increment>0 do
         begin
            If TQ80[J]=true then
            begin
              if K=1 then ComptZ:=80;
              Comptage[K]:=ComptZ;
              Increment:=compt-ComptZ;
              if Increment>80 then Increment:=80;
              ComptZ:=ComptZ+Increment;
              K:=K+1;
            end
            else
            begin
              If Increment=81 then ChercheBlanc;
              Comptage[K]:=ComptZ;
              Increment:=compt-ComptZ;
              if Increment>81 then Increment:=81;
              ComptZ:=ComptZ+Increment;
              K:=K+1;
              end;
         end; end;

Procedure Ecrire1;

begin
      RemplitComptage;
      K:=1;Dep:=1;
      while Comptage[K]>0 do
            begin
            Compt:=Comptage[K];
            Ecrire(Dep);
            Dep:=Comptage[K]+1;
            K:=K+1;
            end; end;




begin clrscr; gotoxy(5,12);writeln('Patience je travaille ');
      for I:=1 to 500 do V[I]:=' ';J:=1;
      assign(P,'FichPara');
      reset(P);
      writeln('J''ai ouvert FichPara');
      Taille:=FileSize(P);writeln('Taille = ',Taille);J:=1;
```

```
      while not EOF(P) do
            begin
            read(P,Zone1);
                J:=J+1;
                end;
                writeln('J''ai rempli les tableaux');

      assign(D,'sunist');
      reset(D);
      assign(A,'arrivee');
      rewrite(A);Num:=1;

            while not EOF(D) do
                  begin
                        Dep:=1;
                        for J:=1 to Taille do
                        begin
                        if TAffNom[J] then
                          begin
                          write(A,TNom[JJ]);
                          CR;
                          end;
                        Compt:=TCompt[J];
                        Lire;
                        if TAffLong[J] then
                          begin
                            if TCompt[J]>80 then
                            begin
                            Ecrire1;Dep:=1;
                            end
                            else Ecrire(Dep);
                            end;
                          end;

                        readln(D);writeln('Je formate la fiche
',Num);
                        Num:=Num+1;
                  end;
                   close(D);
                   close(A);
                   close(P);
                   gotoxy(5,12);write('
');
                   gotoxy(5,12);writeln('C''est terminé, je vais
afficher le fichier formaté ');
                   delay(5000);
                   reset(A);
                   while not EOF(A) do
                   begin
                   read(A,Ch);
                   write (Ch);
                   delay(10);
                   end; end.
```

```
program sunist4b;


type

Zone=record Nom:string[30]; AffNom:boolean; Long:integer;
AffLong:boolean; Q80:boolean; end;



var D,A:text; I,J,Compt,Dep,Num,Taille:integer; V:array [1..500]
of char;Ch:char; TCompt:array[1..50] of integer;TNom:array[1..50]
of string[30]; TAffNom:array[1..50] of
boolean;TAffLong:array[1..50] of boolean; TQ80:array[1..50] of
boolean; Zone1:Zone; P:file of Zone;



{************************   LECTURE
**************************************}

Procedure Lire;{Lit une zone définie dans le fichier à formater}
 begin
           for I:=1 to Compt do
               begin
               read(D,Ch);
               V[I]:=(Ch);
               end;
 end;{Lire}


{************************    ECRITURE
**************************************}


Procedure CR;{Introduit un retour chariot dans le fichier formaté}
begin
     writeln(A,#13); end;{CR}



Procedure Ecrire(var Dep:integer);{Ecrit une zone dans le fichier
formaté,zone de moins de 80 caractères}

begin
            for I:=Dep to Compt do
                begin
                write(A,V[I]);
                end;
                CR; end;{Ecrire}
```

```
Procedure Ecrire1;{Ecrit une zone dans Ie fichier formaté, zone de
plus de 80 caractères}

var

Comptage :array [1..10] of integer; K:integer;


Procedure RemplitComptage;{Remplit un tableau d'entiers des
positions ou l'on devra introduire des retours chariots}

var ComptZ,Increment:integer;


Procedure ChercheBlanc;{Cherche les positions des espaces afin de
ne pas couper des mots par un retour chariot}

begin
     repeat
     ComptZ:=ComptZ-1;
     until V[ComptZ+1]=#32; end;{Chercheb1anc}

Procedure RemplitComptageZero;{Remplit le tableau des positions de
coupures de 0}

begin
     for K:=1 to 10 do Comptage[K]:=0; end;{RemplitComptageZero}


begin
     RemplitComptageZero;
     Increment:=81;ComptZ:=81;K:=1;
     while Increment>0 do
        begin
           If TQ80[J]=true then
           begin
             if K=1 then ComptZ:=80;
             Comptage[K]:=ComptZ;
             Increment:=compt-ComptZ;
             if Increment>80 then Increment:=80;
             ComptZ:=ComptZ+Increment;
             K:=K+1;
           end
           else
           begin
             If Increment=81 then ChercheBlanc;
             Comptage[K]:=ComptZ;
             Increment:=compt-ComptZ;
             if Increment>81 then Increment:=81;
             ComptZ:=ComptZ+Increment;
             K:=K+1;
             end;
        end; end;{RemplitComptage}

begin
     RemplitComptage;
     K:=1;Dep:=1;
```

```
      while Comptage[K]>0 do
            begin
            Compt:=Comptage[K];
            Ecrire(Dep);
            Dep:=Comptage[K]+1;
            K:=K+1;
            end; end;{Ecrire1}


{**************************  PROGRAMME
************************************}


begin {Début du programme} clrscr; gotoxy(5,12);writeln('Patience
je travaille ');
     for I:=1 to 500 do V[I]:=' ';
     assign(P,'FichPara');{Ouverture du fichier des paramètres de
formatage}
     reset(P);
     writeln('J''ai ouvert FichPara');
     Taille:=FileSize(P);writeln('Taille = ',Taille);
     J:=1;
     while not EOF(P) do
         begin
          read(P,Zone1);
           with Zone1 do
                begin
                   TNom[J]:=Nom;
                   TAffNom[J]:=AffNom;
                   TCompt[J]:=Long;
                   TAffLong[J]:=AffLong;
                   TQ80[J]:=Q80;
                end;
           J:=J+1;
          end;
     writeln('J''ai rempli les tableaux');
     assign(D,'sunist');
     reset(D);
     assign(A,'arrivee');
     rewrite(A);
           Num:=1;
           while not EOF(D) do
                 begin
                      Dep:=1;
                      for J:=1 to Taille do
                      begin
                          if TAffNom[J] then
                           begin
                             write(A,TNom[J]);
                             CR;
                           end;
                        Compt:=TCompt[J];
                        Lire;
                        if TAffLong[J] then
                           begin
                             if TCompt[J]>80 then
                               begin
```

```
                                    Ecrire1;Dep:=1;
                                   end
                                 else Ecrire(Dep);
                               end;
                          end;
                        readln(D);writeln('Je formate la fiche
',Num);
                        Num:=Num+1;
                  end;
                  close(D);
                  close(A);
                  close(P);
                  gotoxy(5,12);write('
');
                  gotoxy(5,12);writeln('C''est terminé, je vais
afficher le fichier formaté ');
                  delay(5000);
                  reset(A);
                  while not EOF(A) do
                  begin
                  read(A,Ch);
                  write (Ch);
                { delay(10);}{Ces parenthèses peuvent être
enlevées pour voir défiler le fichier plus lentement}
        end; end.{Fin du programme}
```

```
Programme sunlog.inc (* ensemble de routines utiles au programme
sunlog.pas*)

(* ========== ROUTINES GENERALES DE CONVERSION =======*)

 function EntVersChaine(Num, Largeur : integer ): UneChaine; {
Change un entier en une chaine } var ChaineTemp : UneChaine; begin
  Str(Num:Largeur, ChaineTemp);
  EntVersChaine := ChaineTemp; end; { EntVersChaine }

function EntVersCh0(Num, Largeur : integer) : UneChaine; { Change
un entier en une chaine et ajoute un zéro sur la gauche si
  l'entier est inférieur à 10 } begin
  if Num < 10 then
    EntVersCh0 := '0' + EntVersChaine(Num, Largeur)
  else
    EntVersCh0 := EntVersChaine(Num, Largeur); end; { EntVersCh0}

function ReelEnChaine(Num : real; Largeur, Places : integer) :
UneChaine; { Change un nombre réel en une chaine } var ChaineTemp
: UneChaine; begin
  Str(Num:Largeur:Places, ChaineTemp);
  ReelEnChaine := ChaineTemp; end; { ReelEnChaine }

{ ================== ROUTINES GENERALES  POUR LES CHAINES
=================== }

function ChaineFixe(FString : UneChaine; Len : byte) : UneChaine;
{ Donne une longueur spécifiée à une chaine. Si elle est trop
longue, les
  caractères excédentaires seront tronqués. Si elle est trop
courte, elle
  sera complétée par des blancs. } var LongChaine : byte absolute
FString;
                    { Transforme en variable la longueur de la
chaine } begin
  if LongChaine > Len then
    Delete(FString, Succ(Len), LongChaine - Len)
                    { Supprime la fin d'une chaine trop longue }
  else
    while LongChaine < Len do     { Complète avec des espaces à
droite   }
      FString := FString + ' ';
  ChaineFixe := FString; end; { ChaineFixe }

function Majuscule(S : UneChaine) : UneChaine; { Convertit en
majuscules tous les caractères d'une chaine } var I : integer;
begin                                 { Nous modifions
intentionnellement un   }
  for I := 1 to Length(S) do          { paramétre par valeur, et
retournons    }
    S[I] := UpCase(S[I]);             { celui-ci par le nom de la
fonction     }
  Majuscule := S; end; { Majuscule }
```

```
function CentrerChaine(S : UneChaine) : UneChaine; { Centre une
chaine en trouvant la différence entre la longueur de la chaine
  et celle d'une ligne puis en ajoutant moitié de la différence au
début de
  la chaine. } var Compteur : byte; begin
  for Compteur := 1 to (80 - Length(S)) DIV 2 do
    S := ' ' + S;
  CentrerChaine := S; end; { CentrerChaine }



{ ==================== ROUTINES DE GESTION DE L'ECRAN
====================== }

function ModeMono : boolean; { Détermine si l'ordinateur est en
mode mono ou couleur. Prend un paramètre de la ligne de commande.
Les paramètres B, C, et M fixent respectivement les modes BW80,
CO80 et mono. Si aucun paramètre n'est fourni, l'interruption $10
est utilisée pour obtenir le mode vidéo. } var
  Reg : Registres;    { Registres utilisés pour les appels DOS et
BIOS }
  Ch : char; begin
  if ParamCount > 0 then          { Vérifie si paramètre sur ligne
de commande }
    Ch := Copy(ParamStr(1), 1, 1)
  else
    Ch := ' ';                                          { Pas de
paramètres   }
  case UpCase(Ch) of
    'B', 'M' : ModeMono := True;
    'C' :      ModeMono := False;
    else
    begin
      with Reg do               { Apeller Inter.$10 pour connaître
mode video }
      begin
        AH := $0F;
        Intr($10, Reg);
        ModeMono := (AL <> $03);
      end; { with }
    end;
  end; { case } end; { ModeMono }

procedure FixeCouleur(NewColor : TypeCouleur); { Fixe les couleurs
du fond et des caractéres } begin
  with Couleurs[NewColor] do
  begin
    TextColor(Caract);
    TextBackground(Fond);
  end; { with } end; { FixeCouleur }

procedure FixeFenetre(Fen : FenetreEnreg); { Change la fenêtre
courante pour une fenêtre particulière, déplace le curseur
  dans la fenêtre et "se souvient" de la fenêtre courante dans la
variable
  FenCour. } begin
  with Fen do
```

```
  begin
    FixeCouleur(Couleur);
    Window(Col, Ligne, Pred(Col + Largeur), Pred(Ligne +
Hauteur));
    GotoXY(1, 1);
  end; { with }
  FenCour := Fen; end; { FixeFenetre }

procedure DessinCadre(Fen : FenetreEnreg); { Trace un cadre en
couleur autour d'une fenêtre } type
  CadreEnreg = record
    UL, UR, LL, LR, Horiz, Vert : char;
  end; const
  Cadres : array[CdreMince..CdreLarge] of CadreEnreg =
    ((UL : '┌'; UR : '┐'; LL : '└';                    { CdreMince
}
      LR : '┘'; Horiz  : '─'; Vert : '│'),
     (UL : '╒'; UR : '╕'; LL : '╘';                    { CdreLarge
}
      LR : '╛'; Horiz  : '═'; Vert : '│')); var Compteur : byte;
begin
  with Fen do
  begin
    FixeFenetre(FenPleinEcran);
    FixeCouleur(Couleur);
    GotoXY(Col, Ligne);
    Write(Cadres[Cadre].UL);                          { Trace le
cadre }
    for Compteur := 1 to (Largeur - 2) do
      Write(Cadres[Cadre].Horiz);
    Write(Cadres[Cadre].UR);
    for Compteur := 1 to (Hauteur - 2) do
    begin
      GotoXY(Col, Ligne + Compteur);
      Write(Cadres[Cadre].Vert);
      GotoXY(Pred(Col + Largeur), WhereY);
      Write(Cadres[Cadre].Vert);
    end;
    GotoXY(Col, Pred(Ligne + Hauteur));
    Write(Cadres[Cadre].LL);
    for Compteur := 1 to (Largeur - 2) do
      Write(Cadres[Cadre].Horiz);
    Write(Cadres[Cadre].LR);
  end; { with }
  FixeFenetre(Fen); end; { DessinCadre }

procedure EffaceFen(Fen : FenetreEnreg); { Efface une fenêtre
particulière, laissant le cadre intact, et définit
  cette fenêtre comme la fenêtre courante. } begin
  with Fen do
  begin
    FixeCouleur(DefaultColor);
    Window(Succ(Col), Succ(Ligne), (Col + Largeur - 2), (Ligne +
Hauteur - 2));
    GotoXY(1, 1);
    ClrScr;
  end;
```

```
  FixeFenetre(Fen); end; { EffaceFen }

procedure CreeFenetre(Fen : FenetreEnreg); { Crée une nouvelle
fenêtre - trace un cadre autour de la fenêtre si la
  définition de la fenêtre le spécifie. } begin
  with Fen do
  begin
    FixeFenetre(Fen);
    ClrScr;
    if Cadre <> SansCadre then
      DessinCadre(Fen);
  end; { with } end; { CreeFenetre }

procedure DetruitFen(Fen : FenetreEnreg); { Efface une fenêtre de
l'écran } begin
  with Fen do
  begin
    FixeFenetre(Fen);
    FixeCouleur(DefaultColor);
    ClrScr;
  end; { with } end; { DetruitFen }

procedure ChangeCurseur(Allumer : boolean); { Utilise
l'interruption $10 pour allumer ou éteindre le curseur } const
  AncScanDeb : byte = 0;      { Sauvegarde la valeur des lignes du
curseur }
  AncScanFin : byte = 0; var Reg : Registres;          { Registres
pour les appels au DOS et au BIOS    } begin
  with Reg do
  begin
    case Allumer of
      ON : begin
             AH := 1;
             CH := AncScanDeb;
             CL := AncScanFin;
             Intr($10, Reg);                          { Curseur allumé
}
           end;
     OFF : begin
             AH := 3;
             BH := 0;
             Intr($10, Reg);
             AncScanDeb := CH; { Sauvegarde la valeur des lignes
du curseur }
             AncScanFin := CL;
             AH := 1;
             CH := 32;
             CL := 0;
             Intr($10, Reg);                          { Curseur éteint
}
           end;
    end; { case }
  end; { with } end; { ChangeCurseur }

procedure Affiche(S : UneChaine; PColor : TypeCouleur); { Ecrit
une chaine avec la couleur spécifiée } begin
  FixeCouleur(PColor);
```

```
  Write(S); end; { Affiche}

procedure AfficheXY(S : UneChaine; PColor : TypeCouleur; Col,
Ligne : byte); { Ecrit une chaine à la position et avec la couleur
spécifiées } begin
  GotoXY(Col, Ligne);
  Affiche(S, PColor); end; { AfficheXY }

procedure AfficheChMenuXY(Pstring : UneChaine; Col, Ligne : byte);
{ Ecrit une chaine de commande du menu principal; met toute
capitale de la
  chaine en video haute intensité. } var Compteur : byte; begin
  GotoXY(Col, Ligne);
  for Compteur := 1 to Length(Pstring) do
  begin
    if PString[Compteur] in ['A'..'Z'] then
      Affiche(PString[Compteur], MenuHiColor)
    else
      Affiche(PString[Compteur], MenuLoColor);
  end; end; { AfficheChMenuXY }

procedure AfficheAide(Touche, Explication : UneChaine); { Affiche
les touches d'aide } begin
  Affiche(Touche, HelpKeyColor);
  Affiche(Explication, HelpMessageColor); end; { AfficheAide }

procedure AfficheMenu; { Affiche les différentes options à l'écran
} begin
  CreeFenetre(FenMenu);
  AfficheChMenuXY('mEttre fichier au format', 3, 2);
{ Colonne gauche }
  AfficheChMenuXY('creer une analySe', 3, 3);
  AfficheChMenuXY('Liste des fichiers', 3, 4);
  AfficheChMenuXY('afficher un Fichier', 3, 5);
  AfficheChMenuXY('ESCape', 24, 6);              { Centré dans le
menu } end; { AfficheMenu }



procedure AfficheDirCour; { Affiche le répertoire en cours en haut
de l'écran } var S : UneChaine; begin
  GetDir(0, S);
  FixeFenetre(FenDirCour);
  AfficheXY(ChaineFixe(S, 66), CurrDirColor, 1, 1); end; {
AfficheDirCour }



{ ============================ ROUTINES D'ENTREE
========================== } procedure Arret(Msg : UneChaine);
forward; { Arrête le programme et affiche un message }

function LitCaract(AllumerCurseur : boolean) : char; { Lit un
caractère, convertit les touches du curseur ou de fonction en
ASCII,
```

```
  convertit toutes les lettres en majuscule. Les touches de
fonction et du
  curseur génèrent deux codes - le caractère null suivi d'un code
ASCII bas.
  Turbo convertit le null en ESC; cette routine met à 1 le 8ième
bit du
  second caractère en ajoutant 128 à sa valeur ASCII. Un
identificateur est
  déclaré au début du programme pour chaque touche spéciale (F1,
F2, etc.).
  Pendant l'attente de la frappe au clavier, l'heure est mise à
jour. } var Ch : char; begin
  if AllumerCurseur then                          { Allume le curseur
si demandé   }
    ChangeCurseur(ON);
  {repeat
  until KeyPressed;}
  Read(Kbd, Ch);
  if (Ch = ESC) and KeyPressed then       { Lit le 2ième caractère
}
  begin
    Read(Kbd, Ch);
    Ch := Chr(Ord(Ch) + 128);             { 8 ième bit à 1    }
  end;
  if Ch = ^C then                         { Arrêt sur Contrôle-C }
    Arret('** INTERRUPTION **');
  LitCaract := UpCase(Ch);                { Convertit en majuscule
}
  if AllumerCurseur then                  { Eteint le curseur s'il
a étét allumé }
    ChangeCurseur(OFF); end; { LitCaract }

function LitCaractValide(LegalSet : EnsCaract; CurseurOn :
boolean) : char; { Lit les caractères au clavier jusqu'à ce que le
caractère entré soit dans
  l'ensemble spécifié. } var Ch : char; begin
  repeat
    Ch := LitCaract(CurseurOn);
  until Ch in LegalSet;
  LitCaractValide := Ch; end; { LitCaractValide }

function Confirmer(Msg : UneChaine) : char; { Affiche une question
ou un avertissement et retourne la réponse,
  'O' ou 'N, de l'utilisateur. } begin
  AfficheXY(Msg + ' (O/N)? ', ErrorColor, 1, 5);
  Confirmer := LitCaractValide(['O', 'N'], ON); end; { Confirmer }

function LitChaine(Len : byte) : UneChaine; { Lit une chaine au
clavier }

procedure EffaceEntree(Len, ColUn, LigneUn : byte); { Efface la
chaine d'entrée sur l'écran } begin
  AfficheXY(ChaineFixe('', Len), DefaultColor, ColUn, LigneUn);
                                          { Efface la chaine }
end; { EffaceEntree }

function TraiteEntree(MaxLen, ColUn, LigneUn : byte;
```

```
                                        var ChaineEntree : UneChaine) :
char; { Lit et retourne la touche pressée et modifie la chaine en
entrée } var
  Inp : char;
  LongCour : byte absolute ChaineEntree; begin
  Inp := LitCaractValide([#32..#126, BS, CR, ^A, ^U, ^X, ESC],
ON);
  FixeCouleur(DefaultColor);
  case Inp of
    #32..#126 : if LongCour < MaxLen then    { Ajoute le nouveau
caractère }
      begin
        ChaineEntree := ChaineEntree + Inp;
        Write(Inp);
      end;
    BS : if LongCour > 0 then                { Recule d'un
caractère }
      begin
        Write(BS + ' ' + BS);
        Delete(ChaineEntree, LongCour, 1);
      end;
    ^A, ^U, ^X : begin                       { Efface la chaine
en entrée }
      EffaceEntree(LongCour, ColUn, LigneUn);
      ChaineEntree := '';
      GotoXY(ColUn, LigneUn);
    end;
    ESC : begin
      EffaceEntree(LongCour, ColUn, LigneUn);
      ChaineEntree := ESC;                    { Termine
l'entrée }
    end;
  end; { case }
  TraiteEntree := Inp; end; { TraiteEntree }

var
  Strng : UneChaine;
  PremierX, PremierY : byte;
  Ch : char; begin { LitChaine }
  Strng := '';
  PremierX := WhereX;  { Sauvegarde l'emplacement du curseur avant
toute entrée }
  PremierY := WhereY;
  repeat
    Ch := TraiteEntree(Len, PremierX, PremierY, Strng);
  until Ch in [ESC, CR];
  if Strng <> ESC then
    AfficheXY(Majuscule(Strng), DefaultColor, PremierX, PremierY);
                                    { Re-affiche la chaine en
majuscule }
  LitChaine := Majuscule(Strng); end; { LitChaine }

procedure LitOptionMenu(var Op : OpType); { Lit une option au
clavier - convertit l'entrée en un type d'option } var Ch : char;
begin
  GotoXY(1, 14);
```

```
  Ch := LitCaractValide(['L', 'C', 'E', 'R', 'F', 'A', 'T', 'S',
ESC], OFF);
  case Ch of
    'L': Op := DirOp;
    'C': Op := CopieOp;
    'E': Op := ESunOp;
    'R': Op := RenommeOp;   { Convertit les caractères en une
opération }
    'F': Op := TypeOp;
    'A': Op := LogOp;
    'T': Op := CreeOp;
    'S': Op := AnaOp;
    ESC: Op := EscapeOp;
  end; { case } end; { LitOptionMenu }

{ ============================ ROUTINES  GENERALES
============================ } procedure Beep; { Produit un son plus
agréable que ^G } begin
  Sound(220);
  Delay(100);
  NoSound; end; { Beep }



procedure Message(Msg : UneChaine); { Affiche un message au bas de
l'écran - sauvegarde les valeurs de
  l'ancienne fenêtre, efface la ligne des messages, écrit le
message,
  attend que l'on presse la touche ESC, efface le message d'erreur
puis
  restaure l'ancienne fenêtre. } var
  AncFen : FenetreEnreg;
  Ch : char; begin
  AncFen := FenCour;
  FixeFenetre(FenMessage);
  GotoXY(1, 2);     { Efface la ligne de messages avec la couleur
correcte }
  FixeCouleur(ErrorColor);
  ClrEol;
  AfficheXY(ChaineFixe(CentrerChaine(Msg + '.  Taper ESC pour
continuer.'), 79),
            ErrorColor, 1, 2);
  Ch := LitCaractValide([ESC], OFF);            { Attend la frappe
de ESC }
  FixeCouleur(DefaultColor);                    { Efface le message
d'erreur }
  GotoXY(1, 2);
  ClrEol;
  FixeFenetre(AncFen); end; { Message }

function ErreurES(NomFichier : UneChaine; Op : OpType) : boolean;
{ Vérifie IOResult et affiche un message si une erreur est
survenue.
  Utilise la variable Op pour "connaître" quelle opération était
en cours. } var
  Msg : UneChaine;
  Resultat : integer; begin
```

```
  Resultat := IOResult;
  if Resultat <> 0 then
  begin
    NomFichier := '"' + NomFichier + '"';     { Met le nom de
fichier entre }
    case Resultat of                          { apostrophes }
     $01  : case Op of
              CopieOp, TypeOp, ESunOp, LogOp
                      : Msg := NomFichier + 'non trouvé ';
              RenommeOp : Msg := 'Le fichier ' + NomFichier + '
existe déjà';
              CreeOp    : Msg := 'Le répertoire ' + NomFichier + '
ne peut pas être créé';
              AnaOp     : Msg :=  NomFichier + ' ne peut pas être
supprimé';
            end; { case }
      $FO : Msg :=  NomFichier + ' ne peut pas être copié' +
'--DISQUE PLEIN';
      else
        Msg := 'Erreur d''E/S #' + EntVersChaine(Resultat, 1);
    end; { case }
    Beep;
    Message(Msg);
    ErreurES := True;
  end
  else
    ErreurES := False; end; { ErreurES }


{======================  ROUTINES DE SUNIST4B
============================}


procedure sunist4b(var Fichier1,Fichier2:UneChaine);

(* Ici on retrouve le premier programme entièrement mis au point
*)
(* Maintenant la procédure demande deux noms de fichier pour
travailler , un nom de fichier à analyser; un nom de fichier de
paramètres *)
 type

Zone=record Nom:string[30]; AffNom:boolean; Long:integer;
AffLong:boolean; Q80:boolean; end;



var D,A:text; I,J,Compt,Dep,Num,Taille:integer; V:array [1..500]
of char;Ch:char; TCompt:array[1..50] of integer;TNom:array[1..50]
of string[30]; TAffNom:array[1..50] of
boolean;TAffLong:array[1..50] of boolean; TQ80:array[1..50] of
boolean; Zone1:Zone; P:file of Zone;
```

```
Procedure Lire;{Lit une zone définie dans le fichier à formater}
 begin
          for I:=1 to Compt do
              begin
              read(D,Ch);
              V[I]:=(Ch);
              end;
 end;{Lire}


Procedure CR;{Introduit un retour chariot dans le fichier formaté}
begin
     writeln(A,#13); end;{CR}


Procedure Ecrire(var Dep:integer);{Ecrit une zone dans le fichier
formaté,zone de moins de 80 caractères}

begin
             for I:=Dep to Compt do
                 begin
                 write(A,V[I]);
                 end;
                 CR; end;{Ecrire}


Procedure Ecrire1;{Ecrit une zone dans le fichier formaté, zone de
plus de 80 caractères}

var

Comptage :array [1..10] of integer; K:integer;


Procedure RemplitComptage;{Remplit un tableau d'entiers des
positions ou l'on devra introduire des retours chariots}

var ComptZ,Increment:integer;


Procedure ChercheBlanc;{Cherche les positions des espaces afin de
ne pas couper des mots par un retour chariot}

begin
     repeat
     ComptZ:=ComptZ-1;
     until V[ComptZ+1]=#32; end;{Chercheblanc}

Procedure RemplitComptageZero;{Remplit le tableau des positions de
coupures de 0}

begin
```

```
      for K:=1 to 10 do Comptage[K]:=0; end;{RemplitComptageZero}


begin
      RemplitComptageZero;
      Increment:=81;ComptZ:=81;K:=1;
      while Increment>0 do
         begin
            If TQ80[J]=true then
            begin
              if K=1 then ComptZ:=80;
              Comptage[K]:=ComptZ;
              Increment:=compt-ComptZ;
              if Increment>80 then Increment:=80;
              ComptZ:=ComptZ+Increment;
              K:=K+1;
            end
            else
            begin
              If Increment=81 then ChercheBlanc;
              Comptage[K]:=ComptZ;
              Increment:=compt-ComptZ;
              if Increment>81 then Increment:=81;
              ComptZ:=ComptZ+Increment;
              K:=K+1;
              end;
        end; end;{RemplitComptage}

begin
      RemplitComptage;
      K:=1;Dep:=1;
      while Comptage[K]>0 do
            begin
            Compt:=Comptage[K];
            Ecrire(Dep);
            Dep:=Comptage[K]+1;
            K:=K+1;
            end; end;{Ecrire1}



begin {Début de Sunist4b} clrscr; gotoxy(5,12);writeln('Patience
je travaille ');
      for I:=1 to 500 do V[I]:=' ';
      assign(P,Fichier2);{Ouverture du fichier des paramètres de
formatage}
      reset(P);
      writeln('J''ai ouvert FichPara');
      Taille:=FileSize(P);writeln('Taille = ',Taille);
      J:=1;
      while not EOF(P) do
          begin
           read(P,Zone1);
           with Zone1 do
                begin
                   TNom[J]:=Nom;
                   TAffNom[J]:=AffNom;
```

```
                    TCompt[J]:=Long;
                    TAffLong[J]:=AffLong;
                    TQ80[J]:=Q80;
                  end;
            J:=J+1;
          end;
      writeln('J''ai rempli les tableaux');
      assign(D,Fichier1);
      reset(D);
      writeln('J''ai ouvert ',Fichier1);
      if length(Fichier1)<=8 then Fichier3:=Fichier1+'.ARR'
      else Fichier3:=copy(Fichier1,1,15)+'.ARR';
      assign(A,Fichier3);
      rewrite(A);
            Num:=1;
            while not EOF(D) do
                  begin
                        Dep:=1;
                        for J:=1 to Taille do
                        begin
                            if TAffNom[J] then
                             begin
                               write(A,TNom[J]);
                               CR;
                             end;
                          Compt:=TCompt[J];
                          Lire;
                          if TAffLong[J] then
                             begin
                               if TCompt[J]>80 then
                                 begin
                                  Ecrire1;Dep:=1;
                                 end
                               else Ecrire(Dep);
                             end;
                         end;
                       readln(D);writeln('Je formate la fiche
',Num);
                        Num:=Num+1;
                  end;
                  close(D);
                  close(A);
                  close(P);
                  writeln('C''est terminé, le fichier formaté
s''appelle ',Fichier3);
                  delay(2000); end;{Fin de sunist4b}

{===============   ROUTINES DE CREER UNE ANALYSE
=========================}


Procedure CreerAnalyse;  (* C'est cette procèdure qui va permettre
de crèer de nouveaux fichiers de paramètres qui permettront de
formater des fichiers sur d'autres modèles *)

type
```

```
Zone=record Nom:string[30]; AffNom:boolean; Long:integer;
AffLong:boolean; Q80:boolean; end;

var Num,J:integer;chaine,chaine1:string[30];ch:char;vrai:boolean;
TCompt:array[1..50] of integer;TNom:array[1..50] of string[30];
TAffNom:array[1..50] of boolean;TAffLong:array[1..50] of boolean;
TQ80:array[1..50] of boolean; Zone1:Zone; P:file of Zone;



Procedure AfficheQuestions;

begin
     gotoxy(5,12);writeln('
');
     gotoxy(5,10);writeln('Nom de la zone ',J);
     gotoxy(5,11);writeln('Afficher le nom de la zone  O/N ');
     gotoxy(5,12);writeln('Longueur de la zone ');
     gotoxy(5,13);writeln('Afficher la zone  O/N ');
     gotoxy(5,14);writeln('Chercher les blancs  O/N '); end;

Procedure LireReponses;



begin
     gotoxy(38,10);writeln('                                        ');
     gotoxy(38,11);writeln('                                        ');
     gotoxy(38,12);writeln('                                        ');
     gotoxy(38,13);writeln('                                        ');
     gotoxy(38,14);writeln('                                        ');
     buflen:=30;
     gotoxy(38,10);
     read(Chaine);
     TNom[J]:=Chaine;
     buflen:=1;
     gotoxy(38,11);
     {ch:=LitCaractValide (['O','N','o','n'],on);
     ChangeCurseur(ON);}
     read(ch);
     if (ch='O') or (ch='o') then TAffNom[J]:=true
     else TAffNom[J]:=false;
     buflen:=2;
     gotoxy(38,12);
     read(Num);
     TCompt[J]:=Num;
     buflen:=1;
     gotoxy(38,13);
     read(ch);
     if (ch='O') or (ch='o') then TAffLong[J]:=true
     else TAffLong[J]:=false;
     buflen:=1;
     gotoxy(38,14);
     read(ch);
     if (ch='O') or (ch='o') then TQ80[J]:=true
     else TQ80[J]:=false; end;
```

```
Procedure RemplirTableaux;

begin
     gotoxy(5,12);writeln('Nom de votre analyse ');
     gotoxy(5,13);writeln('8 lettres maximum ');
     buflen:=8;gotoxy(28,12);read(chaine1);
     Chaine1:=Chaine1+'.ANA';
     gotoxy(5,12);writeln('
');
     gotoxy(5,13);writeln('
');
     gotoxy(5,12);writeln('Nombre de zones de votre fichier ');
     buflen:=2;gotoxy(40,12);read(Num);
     for J:=1 to Num do
     begin
          AfficheQuestions;
          LireReponses;
     end;
     gotoxy(5,24);writeln('
');
     gotoxy(5,24);
     Writeln('Le Fichier de paramètres s''appelle ',Chaine1); end;



begin
     DetruitFen(FenPrompt);
     DetruitFen(FenMenu);
     CreeFenetre(FenPleinEcran);
     ChangeCurseur(ON);
     RemplirTableaux;
     assign(P,Chaine1);
     rewrite(P);
     gotoxy(5,24);writeln('
');
     gotoxy(5,24);
     writeln('J''ai ouvert ',Chaine1); delay(1000);
     for J:=1 to Num do
         begin
         with Zone1 do
         begin
         Nom:=TNom[J];
         AffNom:=TAffNom[J];
         Long:=TCompt[J];
         AffLong:=TAffLong[J];
         Q80:=TQ80[J];
         end;
         write(P,Zone1);
         end;
     close(P);
     gotoxy(5,24);writeln('
');
     gotoxy(5,24);writeln('Nom du fichier de paramètres crèé:
',Chaine1);delay(1000);
     DetruitFen(FenPleinEcran);
     ChangeCurseur(OFF);
     AfficheMenu; end;
```

```
{ ============= ROUTINES DE GESTION DES FICHIERS ET REPERTOIRES
=============
  Les étapes pour lire un répertoire d'un disque sont:

  1. Créer la zone de tranfert de disque (DTA). La DTA sert aux
routines
     de lecture des répertoires pour garder les informations sur
le fichier
     venant d'être lu. La DTA doit être longue de 43 octets.

  2. Lire la première entrée du répertoire. Utiliser la fonction
DOS $4E
     pour cela. La première information à fournir est le nom
du(des) fichier(s)
     à rechercher. Ce nom peut comprendre des caractères
génériques, permettant
     de rechercher plusieurs fichiers. Les caractères génériques
sont '?' qui
     remplace tout caractère et '*' qui remplace toute chaine de
caractères.
     Par exemple, '*.*' recherchera tous les fichiers, et '*.PAS'
recherchera
     tous les fichiers ayant l'extension '.PAS'. Les informations
retournées
     dans la DTA sont: le nom et l'attribut du fichier, la date et
l'heure de
     la dernière modification et la taille du fichier.

  3. Lire la seconde entrée et les suivantes du répertoire.
Utiliser la fonction
     $4F pour cela. Cette fonction retourne les mêmes informations
que la
     précédente fonction. }

function FicExiste(NomFichier : UneChaine) : boolean; { Vérifie si
un fichier existe en essayant de l'ouvrir. Si le fichier existe,
  il est refermé pour limiter le nombre de fichiers ouverts en
même temps. } var F : file; begin
  Assign(F, NomFichier);
  Reset(F);
  If IOResult = 0 then                                  { Si le fichier
existe..}
  begin
    Close(F);                                           { Fermer le
fichier }
    FicExiste := True;
  end
  else
    FicExiste := False; end; { FicExiste }

procedure FixeDTA; { Place la zone de transfert du DOS à
l'emplacement de la variable globale DTA } var Reg : Registres;
{ Registres utilisés pour les appels DOS et BIOS } begin
  with Reg do
  begin
    AH := $1A;                  { Utilise la fonction $1A pour fixer
la DTA }
```

```
    DS := Seg(DTA);
    DX := Ofs(DTA);
  end; { with }
  MsDos(Reg); end; { FixeDTA }

function LireFichier(Reg : Registres;
        var Attribut : byte) : UneChaine; { Lit le nom et
l'attribut du fichier depuis le disque } var
  P : byte;
  NomFichier : UneChaine; begin
  MsDos (Reg);                        { Appel DOS pour obtenir les
informations }
  if Reg.AX = 0 then      { Le registre AX retourne un nombre <> de
zéro s'il}
  begin                                          { n'y a plus de
fichiers à lire }
    NomFichier[0] := #13;                 { Prendre le nom du
fichier dans DTA }
    Move (DTA[31], NomFichier[1], 13);
    P := Pos (#0, NomFichier);
    Delete (NomFichier, P, 14 - P);
    Attribut := DTA[22];                                  { Attribut
du fichier }
    LireFichier := NomFichier;
  end
  else
  begin
    Attribut := 0;
    LireFichier := '';
  end; end; { LireFichier }

function PremierFichier (FichierALire : UneChaine;
                 var Attribut : byte) : UneChaine; { Lit le
premier fichier dans un répertoire } const
  TousFichiers = 16;          { Registres utilisés pour les appels
DOS et BIOS } var Reg : Registres; begin
  FixeDTA;
  FichierALire := FichierALire + #0;
       { Place #0 à la fin de FichierALire pour créer une
chaine ASCIIZ }
  with Reg do
  begin
    AH := $4E;
    CX := TousFichiers;   { Lire tous les types de fichiers dans
le répertoire }
    DS := Seg(FichierALire);
    DX := Ofs(FichierALire[1]);   { DX pointera sur le premier
caractère de
                                  FichierALire - en utilisant
l'offset de
                                  FichierALire au lieu de
FichierALire[1], la
                                  routine ne marcherait pas car le
DOS
                                  prendrait l'octet de longueur de
la chaine
```

```
                                    comme premier caractère de
FichierALire      }
  end; { with }
  PremierFichier := LireFichier(Reg, Attribut); end; {
PremierFichier }

function FichierSuivant(var Attribut : byte) : UneChaine; { Lit le
second fichier dans un répertoire et les suivants } var Reg :
Registres;      { Registres utilisés pour les appels DOS et BIOS }
begin
  Reg.AH := $4F;
  FichierSuivant := LireFichier(Reg, Attribut); end; {
FichierSuivant }



{ ================= ROUTINES POUR L'AFFICHAGE DE FICHIERS
================= } procedure AfficheFichier(NomFichier :
UneChaine); { Imprime un fichier sur l'écran } var F : text;

procedure AffAideType; { Affiche les touches d'aide au bas de
l'écran } begin
  FixeFenetre(FenMessage);
  FixeCouleur(HelpMessageColor);
  GotoXY(1, 2);
  ClrEol;                                    { Efface tout
message }
  GotoXY(26, 2);
  if EOF(F) then
    AfficheAide('Appuyez sur une touche', ' [ FIN DE FICHHIER ]')
                                    { Message différent à la fin
du fichier  }
  else
  begin
    AfficheAide('ESPACE', '-écran suivant      ');
    AfficheAide('ESC', '-fin');
  end;
  FixeFenetre(FenType); end; { AffAideType }

procedure AfficheEcran(var Ch : char); { Affiche une partie du
fichier } var
  S : UneChaine;
  Lines : byte; begin
  Lines := 0;
  EffaceFen(FenType);
  repeat
    Readln(F, S);
    if Length(S) > 78 then             { Raccourcit le texte pour
la fenêtre }
      Delete(S, 79, Length(S) - 78);
    Lines := Succ(Lines);
    AfficheXY(S, TypeColor, 2, Succ(Lines));
  until (Lines = (FenType.Hauteur - 2)) or (EOF(F));
  if EOF(F) then                       { Afficher les touches d'aide
modifiées }
    AffAideType;
  Ch := LitCaract(OFF); end; { AfficheEcran }
```

```
var Ch : char; begin { AfficheFichier }
  Assign(F, NomFichier);
  Reset(F);
  if ErreurES(NomFichier, TypeOp) then
    Exit;
  CreeFenetre(FenType);
  Ch := ' ';
  AffAideType;
  while (not EOF(F)) and (Ch <> ESC) do
    AfficheEcran(Ch);
  Close(F);
  if ErreurES(NomFichier, TypeOp) then ;
  DetruitFen(FenType);
  DetruitFen(FenMessage); end; { AfficheFichier }

{ =============== ROUTINES DIVERSES POUR FICHIERS ET REPERTOIRES
============ }


procedure RenommerFichier(Source, Dest : UneChaine); { Renomme un
fichier } var F : file; begin
  if not FicExiste(Source) then
    Message('"' + Source + '" inconnu')
  else
  begin
    Assign(F, Source);
    Rename(F, Dest);
    if ErreurES(Dest, RenommeOp) then ;
  end; end; { RenommerFichier }

procedure CreerRep(NouvRep : UneChaine); { Crée un nouveau
répertoire } begin
  MkDir(NouvRep);
  if ErreurES(NouvRep, CreeOp) then ; end; { CreerRep }



procedure ChangerRep(Path : UneChaine); { Change le répertoire
courant } begin
  ChDir(Path);
  if not ErreurES(Path, LogOp) then
    AfficheDirCour; end; { ChangerRep }

{ ==================== ROUTINES D'AFFICHAGE DE REPERTOIRES
================= } type
  EnregRep = record                          { Contient 1 entrée du
répertoire }
    Nom : string[12];
    Attr : byte;
  end;
  PageRep = array[1..ColsDir, 1..LignesDir] of EnregRep;
                                { Tableau des noms et attributs de
fichiers }

function MgrFenetreRep(Op : OpType;
          NomChemin, Masque : UneChaine;
```

```
          GardeX, GardeY : byte) : UneChaine; { Retourne le nom de
fichier choisi par l'utilisateur à l'écran dans une liste
  des fichiers du répertoire sélectionné. On utilise les touches
du curseur
  pour sélectionner un fichier puis on apuie sur RETURN. Si un
répertoire est
  sélectionné et que l'option en cours nécessite un nom de
fichier, les fichiers
  de ce répertoire sont affichés à l'écran et l'utilisateur doit
alors choisir
  un fichier. } type
  DonneesRepertoire = record
                       EntreeCour : byte;    { Numéro du fichier
actuellement
                                             choisi à l'écran
}
                      TotalFichiers : byte;  { Nombre de fichiers
lus dans la page
                                             courante du répertoire
}
                       PageCour : integer;   { Page courante du
répertoire      }
                       DonneesFichier : PageRep; { Info sur nom et
attribut}
                      AutresFichiers : boolean;  { Vrai si on n'est
pas à la fin
                                                   du répertoire
}
                  end; var DirData : DonneesRepertoire;

procedure AfficheEntree(Col, Ligne : byte;
                        DPage : DonneesRepertoire;
                       PColor : TypeCouleur); { Imprime un nom de
fichier à l'écran } begin
  with DPage.DonneesFichier[Col, Ligne] do
  begin
    AfficheXY(Nom, PColor, (14 * Pred(Col)) + 6, Succ(Ligne));
    Affiche(ChaineFixe('', 14 - Length(Nom)), DefaultColor);
  end; { with } end; { AfficheEntree }

function CalcCol(Entree : byte) : byte; { Calcule la colonne
courante basée sur une entrée particulière } begin
  CalcCol := Succ(Pred(Entree) MOD ColsDir); end; { CalcCol }

function CalcLigne(Entree : byte) : byte; { Calcule la ligne
courante basée sur une entrée particulière } begin
  CalcLigne := Pred(Entree + ColsDir) DIV ColsDir; end; {
CalcLigne }

procedure VideoHaute(NouvEntree : byte;
                     var DPage : DonneesRepertoire); { Met une
entrée particulière en video haute intensié et change la valeur
  de EntreeCour } var
  Col, Ligne : byte; begin
  with DPage do
  begin
    Col := CalcCol(EntreeCour);
```

```
    Ligne := CalcLigne(EntreeCour);
    if NouvEntree <> EntreeCour then    { Ancienne entrée en video
normale }
    begin
      if DonneesFichier[Col, Ligne].Attr = SousRepertoire then
        AfficheEntree(Col, Ligne, DPage, SubDirColor)
      else
        AfficheEntree(Col, Ligne, DPage, FileColor);
      EntreeCour := NouvEntree;
      Col := CalcCol(EntreeCour);
      Ligne := CalcLigne(EntreeCour);
    end;
    if DonneesFichier[Col, Ligne].Attr = SousRepertoire then
      AfficheEntree(Col, Ligne, DPage, HiSubDirColor)    { Haute
intensité }
    else
      AfficheEntree(Col, Ligne, DPage, HiFileColor);
  end; { with } end; { VideoHaute }

procedure AffichePageRep(var DPage : DonneesRepertoire); { Affiche
une page de fichiers à l'écran } var
  ColCounter, RowCounter : byte; begin
  with DPage do
  begin
    for RowCounter := 1 to LignesDir do
    begin
      for ColCounter := 1 to ColsDir do
      begin
        with DonneesFichier[ColCounter, RowCounter] do
        begin
          if Attr = SousRepertoire then
            AfficheEntree(ColCounter, RowCounter, DPage,
SubDirColor)
          else
            AfficheEntree(ColCounter, RowCounter, DPage,
FileColor);
        end; { with }
      end;
    end;
    EntreeCour := 1;                        { Commencer par la
1ière entrée }
    VideoHaute(EntreeCour, DPage);
  end; { with } end; { AffichePageRep }

procedure AfficheAideRep; { Affiche les touches d'aide au bas de
l'écran } begin
  FixeFenetre(FenMessage);
  FixeCouleur(HelpMessageColor);
  GotoXY(1, 1);
  ClrEol;
  GotoXY(3, 1);
  AfficheAide(#24, '-');
  AfficheAide(#25, '-sélection suivante  ');
  AfficheAide('PgUp', '-page préc. ');
  AfficheAide('PgDn', '-page suiv. ');
  AfficheAide(#17 + #217, '-choix   ');
  AfficheAide('ESC', '-quitter');
```

```
  FixeFenetre(FenDir); end; { AfficheAideRep }

procedure LirePageRep(var DPage : DonneesRepertoire; PageNouv :
byte); { Lit les fichiers dans une page particulière du répertoire
}

procedure LirePageRepSuiv(var DPage : DonneesRepertoire); { Lit la
page de fichiers suivante en s'assurant que les fichiers ont
  l'attribut corect }

function FichierLegal(NomFichier : UneChaine; Attrib : byte) :
boolean; { Vérifie si un fichier particulier doit apparaître dans
la liste des
  fichiers } begin
  FichierLegal := (((Prompt[Op].Recherche = Tous) or
                ((Prompt[Op].Recherche = FichierSeul) and
                  (Attrib <> SousRepertoire)) or
                ((Prompt[Op].Recherche = DirSeul) and
                  (Attrib = SousRepertoire))) and
                (NomFichier <> '.')) or
              (NomFichier = ''); end; { FichierLegal }

const
  FichierCour : EnregRep = (Nom : ''; Attr : 0);
                    { Contient les infos sur le fichier venant
d'être lues } var
  PremierFicLu : boolean; begin { LirePageRepSuiv }
  PremierFicLu := False;
  with DPage do
  begin
    TotalFichiers := 0;
    repeat
      with DonneesFichier[CalcCol(Succ(TotalFichiers)),
CalcLigne(Succ(TotalFichiers))] do
      begin
        repeat
          if (PageCour = 1) and (TotalFichiers = 0) and not
PremierFicLu then
          begin                    { Lit le premier fichier si
c'est le cas }
            FichierCour.Nom := PremierFichier(NomChemin + Masque,
FichierCour.Attr);
            PremierFicLu := True;
          end;
          Nom := FichierCour.Nom;
          Attr := FichierCour.Attr;
          FichierCour.Nom := FichierSuivant(FichierCour.Attr);
        until FichierLegal(Nom, Attr); {Accepte seulement les
fichiers corrects }
        if Nom <> '' then
          TotalFichiers := Succ(TotalFichiers);
      end; { with }
    until (TotalFichiers = (LignesDir * ColsDir)) or
(FichierCour.Nom = '');
    AutresFichiers := (FichierCour.Nom <> '');
                { Vérifie s'il existe d'autres fichiers pouvant
apparaître }
```

```
  end; { with }                                                    { sur
un écran } end; { LirePageRepSuiv }

const
  MsgLecture = 'Lecture du répertoire..'; var Compteur : byte;
begin { LirePageRep }
  EffaceFen(FenDir);
  AfficheXY(MsgLecture, MessageColor, 30, 5);
  with DPage do
  begin
    FillChar(DonneesFichier, SizeOf(DonneesFichier), 0);
    if PageNouv < PageCour then   { Lit le répertoire jusqu'à la
page courante}
    begin
      PageCour := 1;
      for Compteur := 1 to Pred(PageNouv) do
      begin
        LirePageRepSuiv(DPage);
        PageCour := Succ(PageCour);
      end;
    end;
    PageCour := PageNouv;
    LirePageRepSuiv(DPage);          { Lit la page courante du
répertoire  }
    AfficheXY(ChaineFixe('', Length(MsgLecture)), DefaultColor,
30, 5);
                                                          { Efface le
message  }
    AfficheRepertoire(DPage);
  end; { with } end; { LirePageRep }

function ParcoursRep(var DPage : DonneesRepertoire) : UneChaine; {
Se déplace dans le répertoire en fonction du caractère entré -
retourne
  le nom du fichier sélectionné }

procedure TraiterGauche; { Traitement de la flèche gauche } begin
  with DPage do
  begin
    if EntreeCour = 1 then           { Dernière entrée en haute
intensité  }
      VideoHaute(TotalFichiers, DPage)
    else
      VideoHaute(Pred(EntreeCour), DPage);             { Entrée
précédente }
  end; { with } end; { TraiterGauche }

procedure TraiterDroite; { Traitement de la flèche droite } begin
  with DPage do
  begin
    if EntreeCour = TotalFichiers then {Première entrée en haute
intensité }
      VideoHaute(1, DPage)
    else
      VideoHaute(Succ(EntreeCour), DPage);               { Entrée
suivante }
  end; { with } end; { TraiterDroite }
```

```
procedure TraiterHaut; { Flèche en haut } var Place : integer;
begin
  with DPage do
  begin
    if EntreeCour <= ColsDir then           { Entrée de la ligne
au-dessus }
    begin
      if EntreeCour = 1 then                { Aller sur la
dernière entrée }
        Place := LignesDir * ColsDir
      else
        Place := (Pred(LignesDir) * ColsDir) + Pred(EntreeCour);
                        { Aller sur la ligne du bas de la colonne
précedente }
      while Place > TotalFichiers do
        Place := Place - ColsDir;         { Aller jusqu'à un nom de
fichier }
      if Place = 0 then
        Place := TotalFichiers;
      VideoHaute(Place, DPage);
    end
    else
      VideoHaute(EntreeCour - ColsDir, DPage);        { Monter
d'une ligne }
  end; { with } end; { TraiterHaut }

procedure TraiterBas; { Flèche en bas } var Place : integer; begin
  with DPage do
  begin
    if (EntreeCour + ColsDir) > TotalFichiers then  { Entrée sur
la ligne du bas}
    begin
      if (EntreeCour MOD ColsDir = 0) then      { Aller sur la
1ère entrée }
        Place := 1
      else
        Place := (Pred(EntreeCour) MOD ColsDir) + 2;
                                    { Aller en haut de la colonne
suivante }
      if Place > TotalFichiers then
        Place := 1;
      VideoHaute(Place, DPage);
    end
    else
      VideoHaute(EntreeCour + ColsDir, DPage);
  end; { with } end; { TraiterBas }

function TraiterCR(var Ch : char) : UneChaine; { Traitement de la
touche RETOUR - retourne un nom de fichier si l'un
  était sélectionné } begin
  with DPage do
  begin
    with DonneesFichier[CalcCol(EntreeCour),
CalcLigne(EntreeCour)] do
    begin
      if (Attr <> SousRepertoire) and (Op = DirOp) then
      begin
```

```
        Beep;
        Message('Vous ne pouvez choisir qu''un répertoire ici');
        Ch := ' ';                                  { Aucun fichier
sélectionné }
        TraiterCR := '';
      end
      else if ((Attr = SousRepertoire) and (Prompt[Op].Select =
DirSeul)) or
         (Attr <> SousRepertoire) then
        TraiterCR := NomChemin + Nom
 { Un répertoire, nécessaire pour l'option courante, a été
sélectionné }
      else
      begin              { Lit et affiche les fichiers du
répertoire choisi }
        ChDir(Nom);
        GetDir(0, NomChemin);
        if NomChemin[Length(NomChemin)] <> '\' then
          NomChemin := NomChemin + '\';
        Masque := '*.*';
        FixeFenetre(FenPrompt);
        AfficheXY(ChaineFixe(NomChemin + Masque, 80 - GardeX),
DefaultColor,
                  GardeX, GardeY);
        LirePageRep(DPage, 1);   { Lire la 1ère page du nouveau
répertoire }
        Ch := ' ';                              { Aucun fichier
sélectionné    }
        TraiterCR := '';
      end;
    end; { with }
  end; { with } end; { TraiterCR }

var Ch : char; begin { ParcoursRep }
  with DPage do
  begin
  repeat
    Ch := LitCaract(OFF);
    if TotalFichiers > 0 then
    begin
      case Ch of
        LeftKey  : TraiterGauche;
        RightKey : TraiterDroite;
        UpKey    : TraiterHaut;
        DownKey  : TraiterBas;
        PgUpKey  : if PageCour > 1 then
                     LirePageRep(DPage, Pred(PageCour));
        PgDnKey  : if AutresFichiers then
                     LirePageRep(DPage, Succ(PageCour));
        HomeKey  : VideoHaute(1, DPage);          { Aller sur le
premier  }
        EndKey   : VideoHaute(TotalFichiers, DPage); { Aller sur
le dernier}
        ESC      : ParcoursRep := ESC;
        CR       : ParcoursRep := TraiterCR(Ch);
      end; { case }
    end
```

```
    else
      ParcoursRep := ESC;
  until Ch in [ESC, CR];
  end; { with } end; { ParcoursRep }

var
  PageRepertoire : DonneesRepertoire; begin { MgrFenetreRep }
  CreeFenetre(FenDir);
  PageRepertoire.PageCour := 1;   { Commencer à la 1ère page du
répertoire }
  LirePageRep(PageRepertoire, 1);
  if PageRepertoire.TotalFichiers = 0 then
    AfficheXY('No file(s)', MessageColor, 35, 5);
  AfficheAideRep;
  MgrFenetreRep := ParcoursRep(PageRepertoire);
  DetruitFen(FenDir);
  DetruitFen(FenMessage); end; { MgrFenetreRep }

{ ================ ROUTINES DE LECTURE DES NOMS DE FICHIERS
=============== } function LitNomFichier(PromptMsg : UneChaine;
                      LigneEntree : byte) : UneChaine; { Lit un nom
de fichier qui sera utilisé par l'opération en cours -
  retourne vrai si un nom de fichier a été lu } begin
  LitNomFichier := ESC;
  if PromptMsg <> '' then
  begin
    AfficheXY(PromptMsg, MessageColor, 1, LigneEntree);
    LitNomFichier := LitChaine(55);
  end; end; { LitNomFichier }

function LitNomFichierRep(PromptMsg : UneChaine;
                          LigneEntree : byte;
                                Op : OpType) : UneChaine; { Lit un
nom de fichier - prend le nom dans le répertoire si un nom de
  fichier n'est pas entré - retourne vrai si un nom de fichier a
été lu }

function AjoutNomChemin(Chem : UneChaine) : UneChaine; { Ajoute le
caractère \ à la fin d'un nom de chemin } begin
  if (Chem <> '') and                                    { Pas de \
à la fin }
     (Chem[Length(Chem)] <> '\') then
    Chem := Chem + '\';                                    {
Ajouter un \  }
  AjoutNomChemin := Chem; end; { AjoutNomChemin }

function NetNomChemin(Chem : UneChaine) : UneChaine; { Retourne un
nom de chemin dont le caractère \ a été enlevé } begin
  if (Chem[Length(Chem)] = '\') and       { Le dernier caractère
est '\'   }
     (Chem <> '\') and                        { Ce n'est pas la
racine       }
     not ((Length(Chem) = 3) and (Copy(Chem, 2, 2) = ':\')) then
    Delete(Chem, Length(Chem), 1);                          {
Supprimer le '\' }
  NetNomChemin := Chem; end; { NetNomChemin }
```

```
procedure ExtraitNomChemin(var Chem, NomFichier : UneChaine; Op :
OpType); { Avec un nom de fichier pouvant être précédé d'un
chemin, cette routine
  met la partie correspondant au chemin dans Chem et le nom du
fichier
  dans NomFichier } var Place : byte; begin
  if (Op = LogOp) and (Length(Chem) = 1) and (Chem[1] in
['A'..'Z']) then
    Chem := Chem + ':';                    { Une lettre seule spécifie
un disque }
  Place := Length(Chem);
  while (Place > 0) and not (Chem[Place] in [':', '\']) do
                                           { revenir au dernier
caractère }
    Place := Pred(Place);
  if Place = 0 then                                  { nom seul,
sans chemin }
  begin
    NomFichier := Chem;
    Chem := '';
  end
  else
  begin  { sépare nom et chemin }
    NomFichier := Copy(Chem, Succ(Place), Length(Chem) - Place);
                                             { prendre la partie
arrière  }
    Chem[0] := Chr(Place);                     { prendre la
partie avant  }
  end;
  if Chem[Length(Chem)] = ':' then          { Ajouter nom de chemin
si seul }
    GetDir(Succ(Ord(Chem[1]) - Ord('A')), Chem); { un disque était
spécifié}
  Chem := AjoutNomChemin(Chem); end; { ExtraitNomChemin }

procedure VerifRepGlobal(var Chem, NomFichier : UneChaine; Op :
OpType); { Détermine en fonction de Chem, NomFichier et l'option
courante si le
  programme cherchera un répertoire complet ou non } var
  Attrib : byte;
  TempName : UneChaine; begin
  if (Pos('*', NomFichier) <> 0) or (Pos('?', NomFichier) <> 0)
then
    Exit;                        { Caractères génériques - pas de
modification }
  if NomFichier = '' then
  begin
    if Chem = '' then                { Chercher tout le répertoire
courant }
    begin
      NomFichier := '*.*';
      Exit;
    end
    else                                     { Seul un chemin a été
spécifié }
      Chem := NetNomChemin(Chem);
  end;
```

```
  TempName := PremierFichier(Chem + NomFichier, Attrib);
  if Attrib = SousRepertoire then  { Vérifier si le fichier
spécifié est un   }
  begin                                { sous-répertoire }
    Chem := AjoutNomChemin(Chem) + AjoutNomChemin(NomFichier);  {
Corriger }
    if Op in [LogOp, AnaOp] then                              { le nom
du chemin }
      NomFichier := '' { Seul un chemin est nécessaire pour LogOp
ou AnaOp }
    else
      NomFichier := '*.*';  { Rechercher entièrement le répertoire
spécifié}
  end
  else if (NomFichier = '') and not (Op in [LogOp, AnaOp]) then
    NomFichier := '*.*';   { Rechercher entièrement le répertoire
spécifié }
  Chem := AjoutNomChemin(Chem);  { Mettre un '\' à la fin du nom
de chemin } end; { VerifRepGlobal }

procedure LitNomDansRep(var NomFichier : UneChaine;
                            AncRep : UneChaine;
                            GardeY : byte;
                               Op : OpType); { Prend un nom
de fichier dans le répertoire } var Chem : UneChaine; begin
  Chem := NomFichier;
  ExtraitNomChemin(Chem, NomFichier, Op); { Séparer chenin et nom
de fichier }
  VerifRepGlobal(Chem, NomFichier, Op);   { Ajouter "*.*" au nom
de fichier }
  GetDir(0, AncRep);                      { Sauvegarder le
répertoire en cours }
  ChDir(NetNomChemin(Chem));              { Changer pour le
répertoire spécifié }
  if not ErreurES(AjoutNomChemin(Chem), LogOp) then  { Si le
répertoire existe..}
  begin
    GetDir(0, Chem);                  { Trouver le nom du chemin
courant    }
    if NomFichier <> '' then          { Mettre un '\' à la fin du
nom de chemin  }
      Chem := AjoutNomChemin(Chem);
    AfficheXY(ChaineFixe(Chem + NomFichier, 55), DefaultColor,
            Succ(Length(Prompt[Op].Prompt1)), GardeY);   {
Afficher le nom     }
    if NomFichier = '' then             { Si seul un nom de chelin
est nécessaire}
      Chem := NetNomChemin(Chem);   { le rendre correct
}
    if (Pos('*', NomFichier) <> 0) or (Pos('?', NomFichier) <> 0)
or
       (Op = DirOp) then
    begin                   { Rechercher en utilisant les caractères
génériques }
      NomFichier := MgrFenetreRep(Op, Chem, NomFichier,
                                  Succ(Length(Prompt[Op].Prompt1)),
GardeY);
```

```
      ChDir(AncRep);                                { Revenir au
répertoire de départ }
      if NomFichier = ESC then
        Exit;
      FixeFenetre(FenPrompt);
      if Prompt[Op].Select <> Aucun then
        AfficheXY(ChaineFixe(NomFichier, 55), DefaultColor,  {
Afficher le nom }
                  Succ(Length(Prompt[Op].Prompt1)), GardeY);
    end
    else begin
      ChDir(AncRep);                               { Revenir au répertoire
de départ }
      NomFichier := Chem + NomFichier;     { Retourner nom complet
avec le chemin }
    end;
  end
  else                                              { Le chemin entré est
illégal }
    NomFichier := ESC; end; { LitNomDansRep }

var
  NomFichier : UneChaine;
  GardeY    : byte;
  GardeRep : UneChaine; begin { LitNomFichierRep }
  LitNomFichierRep := ESC;
  GardeY := WhereY;
  NomFichier := LitNomFichier(Prompt[Op].Prompt1, 1);
  if (NomFichier = '') and (Prompt[Op].Recherche = Aucun) then
    NomFichier := ESC;                              { Sortir si rien n'a
été entré }
  if NomFichier = ESC then
    Exit;
  if Prompt[Op].Recherche <> Aucun then
    LitNomDansRep(NomFichier, GardeRep, GardeY, Op);
  LitNomFichierRep := NomFichier; end; { LitNomFichierRep }
```

```
{$C+,I+}                { Ctrl-C  vérifié, Erreurs d'E/S  vérifiées
} program Sunlog;

const
  Version       = '4.b';
  SousRepertoire = 16;                          { L'attribut d'un
sous-répertoire }
  LignesDir       = 8;
  ColsDir       = 5; type
  EnsCaract = set of char;
  UneChaine = string[80];
  Registres = record                        { Utilisé par les appels
MsDos et Intr }
    case byte of
      1 : (AX, BX, CX, DX, BP, SI, DI, DS, ES, Flags : integer);
      2 : (AL, AH, BL, BH, CL, CH, DL, DH : byte);
  end;
  TypeCouleur = (ErrorColor, LogOnColor, CurrDirColor,
                 HelpKeyColor, HelpMessageColor, MessageColor,
MenuColor,
                 MenuHiColor, MenuLoColor, TypeColor, SubDirColor,
FileColor,
                 HiSubDirColor, HiFileColor, DefaultColor);
  OpType = (DirOp, CopieOp, ESunOp, RenommeOp, TypeOp, LogOp,
CreeOp, AnaOp,
            EscapeOp);  { Les différentes opérations possibles }
  SelectType = (Tous, FichierSeul, DirSeul, Aucun); { Le  type
d'entrée à afficher }
                                                  { ou sélectionner
}
  PromptType = array[DirOp..AnaOp] of record    { Contient les
messages et     }
    Prompt1, Prompt2 : string[34];              {
recherche/sélectionne les    }
    Recherche, Select : SelectType;                { types pour
chaque option     }
  end;
  CouleurDef = record
                 Caract, Fond : byte;
               end;
  Cadres = (SansCadre, CdreMince, CdreLarge);    { Cadres autour
des fenêtres }
  FenetreEnreg = record
                  Col, Ligne : byte;             { Dimension des
fenêtres}
                  Hauteur, Largeur : byte;
                  Couleur : TypeCouleur;       { Couleur par défaut
des fenêtres }
                  Cadre : Cadres;              { Quel cadre autour de
la fenêtre ? }
                end;
  ListeCouleur = array[ErrorColor..DefaultColor] of
CouleurDef;{Table des couleurs}

const
```

```
  ON = True;
  OFF = False;
  Couleurs : ListeCouleur = ((Caract : White;  { ErrorColor }  {
Toutes les     }
                           Fond : Red),                          {
couleurs       }
                          (Caract : White;       { LogOnColor }  {
utilisées dans }
                           Fond : Blue),                         {
le programme   }
                          (Caract : Yellow;      { CurrDirColor }
                           Fond : Black),
                          (Caract : Black;       { HelpKeyColor }
                           Fond : LightGray),
                          (Caract : LightGray;   { HelpMessageColor
}
                           Fond : Blue),
                          (Caract : LightCyan;   { MessageColor }
                           Fond : Black),
                          (Caract : Yellow;      { MenuColor }
                           Fond : Cyan),
                          (Caract : Yellow;      { MenuHiColor }
                           Fond : Cyan),
                          (Caract : White;       { MenuLoColor }
                           Fond : Cyan),
                          (Caract : LightGreen;  { TypeColor }
                           Fond : Black),
                          (Caract : Yellow;      { SubDirColor }
                           Fond : Black),
                          (Caract : White;       { FileColor }
                           Fond : Black),
                          (Caract : Yellow;      { HiSubDirColor }
                           Fond : Blue),
                          (Caract : Yellow;       { HiFileColor }
                           Fond : Blue),
                          (Caract : LightGray;   { DefaultColor }
                           Fond : Black));
  CouleursMono : ListeCouleur = ((Caract : White; { ErrorColor } {
Même table que}
                            Fond : Black),                       {
les couleurs, }
                           (Caract : White;     { LogOnColor }  {
mais pour      }
                            Fond : Black),                       {
moniteurs mono et B/W }
                           (Caract : White;     { CurrDirColor }
                            Fond : Black),
                           (Caract : Black;     { HelpKeyColor }
                            Fond : LightGray),
                           (Caract : LightGray; {
HelpMessageColor }
                            Fond : Black),
                           (Caract : White;     { MessageColor }
                            Fond : Black),
                           (Caract : White;     { MenuColor }
                            Fond : Black),
                           (Caract : White;     { MenuHiColor }
```

```
                                Fond : Black),
                               (Caract : LightGray; { MenuLoColor }
                                Fond : Black),
                               (Caract : White;     { TypeColor }
                                Fond : Black),
                               (Caract : White;     { SubDirColor }
                                Fond : Black),
                               (Caract : LightGray; { FileColor }
                                Fond : Black),
                               (Caract : Black;     { HiSubDirColor }
                                Fond : LightGray),
                               (Caract : DarkGray;  { HiFileColor }
                                Fond : LightGray),
                               (Caract : LightGray; { DefaultColor }
                                Fond : Black));
  FenPleinEcran : FenetreEnreg = (Col : 1;          { Les fenêtres
utilisées }
                                  Ligne : 1;          { dans ce
programme }
                                 Hauteur : 25;
                                 Largeur : 80;
                                 Couleur : DefaultColor;
                                   Cadre : SansCadre);
  FenMenu : FenetreEnreg = (Col : 1;          { Fenêtre du menu
principal }
                             Ligne : 2;
                            Hauteur : 7;
                            Largeur : 35;
                            Couleur : MenuColor;
                              Cadre : CdreLarge);
  FenDir : FenetreEnreg = (Col : 1;           { Pour lister les
répertoires  }
                             Ligne : 14;
                           Hauteur : 10;
                           Largeur : 80;
                           Couleur : FileColor;
                             Cadre : CdreLarge);
  FenDirCour : FenetreEnreg = (Col : 1; {Pour afficher le
répertoire courant }
                               Ligne : 1;
                              Hauteur : 2;
                              Largeur : 80;
                              Couleur : CurrDirColor;
                                Cadre : SansCadre);
  FenPrompt : FenetreEnreg = (Col : 1;     { Fenêtre où
l'utilisateur entre   }
                               Ligne : 9;     { les noms de fichiers,
etc.       }
                             Hauteur : 5;
                             Largeur : 80;
                             Couleur : DefaultColor;
                               Cadre : SansCadre);
  FenType : FenetreEnreg = (Col : 1;  { Pour afficher le contenu
d'un fichier }
                             Ligne : 10;
                            Hauteur : 15;
                            Largeur : 80;
```

```
                          Couleur : TypeColor;
                            Cadre : CdreMince);
  FenMessage : FenetreEnreg = (Col : 1;          { Pour les
messages d'erreur }
                            Ligne : 24;
                          Hauteur : 2;
                          Largeur : 80;
                          Couleur : DefaultColor;
                            Cadre : SansCadre);
  FenLogOn : FenetreEnreg = (Col : 1;            { Affiche
l'écran de départ }
                            Ligne : 1;
                          Hauteur : 6;
                          Largeur : 65;
                          Couleur : LogOnColor;
                            Cadre : CdreLarge);


  { Constante avec type contenant des informations sur chaque
opération: }
  Prompt : PromptType = ((Prompt1 : 'Disque:\Répertoire';
{ DirOp }
                           Prompt2 : '';
                         Recherche : Tous;
                            Select : Aucun),
                          (Prompt1 : 'Fichier à copier: ';  {
CopieOp }
                           Prompt2 : 'Copier vers: ';
                         Recherche : Tous;
                            Select : FichierSeul),
                          (Prompt1 : 'Fichier à mettre au format:
'; { ESunOp }
                           Prompt2 : 'Fichier de paramètres';
                         Recherche : Tous;
                            Select : FichierSeul),
                          (Prompt1 : 'Fichier à renommer: '   ;  {
RenommeOp }
                           Prompt2 : 'Renommer en: ';
                         Recherche : Tous;
                            Select : FichierSeul),
                          (Prompt1 : 'Fichier à afficher: ';
{ TypeOp }
                           Prompt2 : '';
                         Recherche : Tous;
                            Select : FichierSeul),
                          (Prompt1 : 'Changer disque/répert.: ';{
LogOp }
                           Prompt2 : '';
                         Recherche : DirSeul;
                            Select : DirSeul),
                          (Prompt1 : 'Créer le répertoire: ';     {
CreeOp }
                           Prompt2 : '';
                         Recherche : Aucun;
                            Select : Aucun),
                          (Prompt1 : 'Crèer une analyse  O ou ESC
';  { AnaOp }
```

```
                          Prompt2 : '';
                        Recherche : DirSeul;
                           Select : DirSeul));

  { Définition de quelques touches importantes : }
  NULL      =   #0;              Bell      =   #7;
  BS        =   #8;              LF        =  #10;
  CR        =  #13;              ESC       =  #27;
  HomeKey   = #199;              EndKey    = #207;  { contrôle du
curseur }
  UpKey     = #200;              DownKey   = #208;
  PgUpKey   = #201;              PgDnKey   = #209;
  LeftKey   = #203;              InsKey    = #210;
  RightKey  = #205;              DelKey    = #211;
  F1        = #187;              F6        = #192;  { touches de
fonction }
  F2        = #188;              F7        = #193;
  F3        = #189;              F8        = #194;
  F4        = #190;              F9        = #195;
  F5        = #191;              F10       = #196;

var
  OpCour : OpType;                    { L'option en cours
d'exécution }
  FenCour : FenetreEnreg;             { Information sur la fenêtre
courante }
  DTA : array[1..43] of byte;
                           { Variable nécessaire pour les transferts
sur disque }
  Fichier3:Unechaine;

{$I LOGIDOCU.INC}

{ ======================ROUTINES D'EXECUTION DU
PROGRAMME==================== } procedure Arret {(Msg :
UneChaine)} ; { Arrête le programme et affiche un message } begin
  CreeFenetre(FenPleinEcran);
  ChangeCurseur(On);
  if Msg <> '' then
    Writeln(Msg);
  Halt; end; { Arret }

procedure AfficheMsgDep; { Affiche le message de départ pendant
1,5 seconde ou jusqu'à la frappe d'une touche } var
  Ch : char;
  Compteur : byte; begin
  CreeFenetre(FenLogOn);
  AfficheXY('LOGICIEL DE CHANGEMENT DE FORMAT DE FICHIER ',
LogOnColor, 3, 2);
  AfficheXY('Version ', LogOnColor, 50, 2);
  AfficheXY(Version, LogOnColor, 60, 2);
  AfficheXY('IBM-PC/XT/AT/PCjr', LogOnColor, 17, 3);
  AfficheXY('Copyright (C) 1988 LOF Inc.', LogOnColor, 3, 5);
  Compteur := 1;
  repeat              { attend 8 secondes ou jusqu'à la frappe d'une
touche }
    Delay(500);
```

```
    Compteur := Succ(Compteur);
  until (Compteur > 15) or KeyPressed;
  while KeyPressed do                              { Vider le buffer du
clavier }
    Ch := LitCaract(OFF);
  DetruitFen(FenLogOn); end; { AfficheMsgDep }

procedure InitEcran; { Initialisation de l'écran, de la date et de
l'heure, vérifier le type
  du moniteur utilisé } begin
  ChangeCurseur(OFF);              { efface le curseur, garde son
emplacement}
  ClrScr;
  if ModeMono then                 { Utiliser les couleurs mono si
mode mono }
    Couleurs := CouleursMono;
  AfficheMsgDep;
  AfficheDirCour;                   { Montrer disque et répertoire
courants  }
  AfficheMenu;                                     { Afficher menu
principal  } end; { InitEcran }

procedure ExecuteOp(Op : OpType); { Routine de contrôle exécutant
l'option demandée dans le menu des
  routines. Certaines opérations nécessitent deux noms de fichiers
  (copier, renommer), d'autres un seul (dir, suppression,etc.) }
var
  Fichier1, Fichier2 : UneChaine;          { Noms des fichiers
temporaires } begin { ExecuteOp }
  { DEMANDER LES NOMS DE FICHIERS: }
  with Prompt[Op] do
  begin
    Fichier1 := LitNomFichierRep(Prompt1, 1, Op);  { Demander 1er
nom de fichier}
    if Fichier1 = ESC then                          { A-t-on taper
ESC? }
      Exit;
    if Prompt[Op].Prompt2 <> '' then
    begin
      Fichier2 := LitNomFichier(Prompt[Op].Prompt2, 3); {Demander
2ième fichier }
      if (Fichier2 = ESC) or (Fichier2 = '') then       {  ESC ou
RETURN? }
        Exit;
    end;
  end; { with }
  { EFFECTUER L'OPERATION DEMANDEE:}     { Commandes DOS
correspondantes }
  case Op of                             {
-------------------------- }
    DirOp      : ;                                  { LitNomFichier
}
    CopieOp    : begin
                   gotoxy(5,12);
                   writeln('Procedure absente ');
                   delay(1000);
                 end;
```

```
    TypeOp     : AfficheFichier(Fichier1);          { type }
    ESunOp     : Sunist4b(Fichier1,Fichier2);          { C'est
cette fonction qu'on a remplacée,Effacefichier(Fichier1),del 0
    RenommeOp  : RenommerFichier(Fichier1, Fichier2); { ren OU
rename }
    CreeOp     : CreerRep(Fichier1);                { md OU mkdir }
    AnaOp      : CreerAnalyse;            { Deuxieme fonction
remplacee }
    LogOp      : ChangerRep(Fichier1);              { cd OU chdir
OU a: }
  end; { case } end; { ExecuteOp }

{ ========================  CORPS DU PROGRAMME
======================== } begin { FileManager }
  InitEcran;                                { Mode video, type du
curseur, etc. }
  LitOptionMenu(OpCour);                    { Acquiert l'entrée de
l'utilisateur}
  while OpCour <> EscapeOp do
  begin
    FixeFenetre(FenPrompt);          { Fenêtre pour les entrées }
    ExecuteOp(OpCour);               { Effectue l'opération
demandée }
    DetruitFen(FenPrompt);           { Efface la fenêtre d'entrée }
    LitOptionMenu(OpCour);           { Demande une entrée à nouveau
}
  end;
  Arret('');                                { Fin du programme; restaure
le curseur }
  changecurseur(ON) end. { FileManager }
```