Historiqie Linux

Premier mail de Linus

TRADUCTION L'histoire de LINUX

```
 Note : Le texte suivant a été écrit par Linus le 31 juillet 1992. Il s'agit d'un
 collection de divers artefacts de la période où Linux est apparu pour la première fois
 a commencé à prendre forme.

 Ce n'est qu'un voyage sentimental dans certains des premiers messages
 concernant Linux, vous pouvez donc appuyer sur « n » maintenant si vous le souhaitez réellement
 Je pensais que tu aurais quelque chose de technique.
```

```
De : torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Groupes de discussion : comp.os.minix
Objet : Gcc-1.40 et une question posix
ID du message :
Date : 3 juil. 91 10:00:50 GMT
```

```
Bonjour les netlanders,
```

```
En raison d'un projet sur lequel je travaille (dans minix), je suis intéressé par le posix
définition standard. Quelqu'un pourrait-il m'indiquer une (de préférence) définition standard
format lisible par machine des dernières règles posix ? Les sites FTP seraient
bon.
```

```
 Le projet était évidemment Linux, donc le 3 juillet j'avais commencé à réfléchir
 à propos des choses réelles au niveau de l'utilisateur : certains pilotes de périphériques étaient prêts,
 et le disque dur fonctionnait réellement. Pas grand chose d'autre.
```

```
En aparté pour tous ceux qui utilisent gcc sur minix - [ supprimé ]
```

```
 Juste un rapport de réussite sur le portage de gcc-1.40 vers minix en utilisant la version 1.37
 version réalisée par Alan W Black & co.
```

```
              Linus Torvalds torvalds@kruuna.helsinki.fi
```

```
PS. Quelqu'un pourrait-il essayer de me contacter depuis l'étranger, car
```

```
j'ai
installé un "plan de changement" (fait par votre serviteur), et je ne suis pas certain
ça marche de l'extérieur ? Il devrait signaler un nouveau .plan à chaque fois.
```

```
 Donc je n'avais aucune idée - je venais juste d'apprendre l'existence des pipes nommées. Poursuivez-moi en justice.
 une partie du message a reçu beaucoup plus de réponses que la requête POSIX réelle,
 mais la requête a attiré ARL hors du bois, et nous avons envoyé des courriers
 pendant un moment, ce qui donne le sous-répertoire Linux sur nic.funet.fi.

 Puis, presque deux mois plus tard, j'ai eu quelque chose qui fonctionnait réellement : je
 les sources de la version 0.01 sont parfois disponibles sur la carte réseau à cette période
 temps. 0.01 les sources n'étaient pas réellement exécutables : elles n'étaient qu'un jeton
 geste à arl qui avait probablement commencé à désespérer de pouvoir un jour y arriver
 n'importe quoi. Ce prochain message doit avoir été publié il y a quelques semaines seulement
 avant cette sortie.
```

```
De : torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Groupes de discussion : comp.os.minix
Objet : Qu'aimeriez-vous voir le plus dans Minix ?
Résumé : petit sondage pour mon nouveau système d'exploitation
ID du message :
Date : 25 août 1991 20:57:08 GMT
Organisation : Université d'Helsinki
```

```
Bonjour à tous ceux qui utilisent Minix -
```

```
Je fais un système d'exploitation (gratuit) (juste un passe-temps, ce ne sera pas grand et
(comme gnu) pour les clones AT 386(486). Cela a été préparé
depuis avril, et commence à se préparer. J'aimerais avoir des retours sur choses que les gens aiment/n'aiment pas dans Minix, car mon système d'exploitation lui ressemble un peu
(même disposition physique du système de fichiers (pour des raisons pratiques)
entre autres).
```

```
J'ai actuellement porté bash (1.08) et gcc (1.40), et les choses semblent fonctionner.
Cela implique que j'obtiendrai quelque chose de pratique dans quelques mois,
```

et
J'aimerais savoir quelles fonctionnalités la plupart des gens
souhaiteraient. Des suggestions
sont les bienvenus, mais je ne promets pas de les mettre en œuvre :-)

Linus (torvalds@kruuna.helsinki.fi)

PS. Oui, il est exempt de tout code minix et dispose d'un système de
fichiers multithread.
Il n'est PAS portable (utilise la commutation de tâches 386, etc.) et il ne
le sera probablement jamais.
prendra en charge tout autre chose que les disques durs AT, car c'est tout
ce que j'ai :-(.

À en juger par la publication, la version 0.01 n'était pas encore sortie,
mais elle n'en était pas loin.
Je suppose que la première version est sortie à la mi-septembre 1991.
j'ai reçu quelques réponses à ce sujet (la plupart par courrier, que je
n'ai pas enregistré), et
J'ai même reçu quelques mails me demandant de devenir bêta-testeur pour
Linux. Après ça
juste quelques réponses générales aux questions sur le net :

De : torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Groupes de discussion : comp.os.minix
Objet : Re : Qu'aimeriez-vous voir le plus dans Minix ?
Résumé : oui, ce n'est pas portable
ID du message :
Date : 26 août 1991 11:06:02 GMT
Organisation : Université d'Helsinki

Dans l'article jkp@cs.HUT.FI (Jyrki Kuoppala) écrit :
>> [re : mon post sur mon nouveau système d'exploitation]
>
>Dis-nous en plus ! Est-ce qu'il faut un MMU ?

Oui, il a besoin d'un MMU (désolé tout le monde), et il a spécifiquement
besoin d'un
386/486 MMU (voir plus loin).

>
>>PS. Oui, il est exempt de tout code Minix et dispose d'un système de
fichiers multithread.
>>Il n'est PAS portable (utilise la commutation de tâches 386, etc.)
>
>Quelle est la part du code en C ? Quelles difficultés rencontrera le
portage ?
>Personne ne vous croira à propos de la non-portabilité ;-), et pour ma
part, je le ferais
>j'aimerais le porter sur mon Amiga (Mach a besoin d'un MMU et Minix n'est

pas gratuit).

Simplement, je dirais que le portage est impossible. C'est principalement en C, mais la plupart
les gens n'appelleraient pas ce que j'écris C. Il utilise toutes les fonctionnalités imaginables
des 386 que j'ai pu trouver, car c'était aussi un projet pour m'apprendre le 386. Comme déjà mentionné, il utilise une MMU, à la fois pour la pagination (pas sur le disque)
encore) et la segmentation. C'est la segmentation qui le rend VRAIMENT 386 dépendant (chaque tâche a un segment de 64 Mo pour le code et les données - max 64 tâches
en 4 Go. Ceux qui ont besoin de plus de 64 Mo par tâche sont des durs à cuire.

Il utilise également toutes les fonctionnalités de gcc que j'ai pu trouver, en particulier __asm__
directive, pour ne pas avoir besoin d'autant d'objets en langage assembleur. Certains de mes fichiers « C » (en particulier mm.c) sont presque autant en assembleur que
C. Il serait même « intéressant » de le porter sur un autre compilateur (bien que
pourquoi quelqu'un voudrait utiliser autre chose que gcc est un mystère).

Remarque : Linux est en fait devenu plus portable avec les versions plus récentes :
il y avait beaucoup plus d'assemblage dans les premières versions. Il a en fait
a été porté sur d'autres architectures maintenant.

Contrairement à Minix, j'aime aussi les interruptions, donc les interruptions sont
traitées sans essayer de cacher la raison derrière elles (j'aime particulièrement
mon pilote de disque dur. Quelqu'un d'autre fait des interruptions pour piloter un état-
machine ?). Dans l'ensemble, c'est le cauchemar des porteurs.

>En ce qui concerne les fonctionnalités ; eh bien, pseudo ttys, sockets BSD, mode utilisateur
>systèmes de fichiers (pour pouvoir dire cat /dev/tcp/kruuna.helsinki.fi/finger),
> taille de la fenêtre dans la structure tty, appels système capables de prendre en charge
>POSIX.1. Oh, et les noms de fichiers longs de style bsd.

La plupart d'entre eux semblent possibles (la structure tty a déjà des stubs pour
taille de la fenêtre), sauf peut-être pour les systèmes de fichiers en mode

utilisateur. Quant à POSIX,
Je serais ravi de l'avoir, mais posix veut de l'argent pour leurs papiers, alors
ce n'est pas une option actuellement. Dans tous les cas, ce sont des choses qui ne seront pas
sera encore pris en charge pendant un certain temps (je vais d'abord en faire un simple minix-
sosie, mot-clé SIMPLE).

Linus (torvalds@kruuna.helsinki.fi)

PS. Pour que les choses soient vraiment claires - oui, je peux exécuter gcc dessus, et bash, et
la plupart des utilitaires gnu [bin/file], mais il n'est pas très débogué, et le
La bibliothèque est vraiment minimale. Elle ne prend même pas encore en charge les disquettes.
ne sera pas prêt à être distribué avant quelques mois. Même alors,
ne sera probablement pas capable de faire beaucoup plus que Minix, et encore moins dans certains
Respectueusement. Ce sera gratuit cependant (probablement sous licence GNU ou similaire).

Eh bien, visiblement quelque chose a fonctionné sur ma machine : je doute que je l'aie déjà fait
j'ai réussi à faire compiler gcc sous Linux (ou j'aurais été trop
j'en suis fier pour ne pas le mentionner). Toujours avant toute date de sortie.

Puis, le 5 octobre, il semble que j'aie publié 0,02. Comme je l'ai déjà
mentionné, 0.01 n'était en fait pas livré avec des binaires : c'était juste
code source pour les personnes intéressées par ce à quoi ressemble Linux. Notez le
absence d'annonce pour 0.01 : je n'en étais pas trop fier, donc je pense que je
j'ai simplement envoyé un message à tous ceux qui avaient manifesté leur intérêt.

De : torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Groupes de discussion : comp.os.minix
Objet : Sources de noyau gratuites de type minix pour 386-AT
ID du message :
Date : 5 oct. 91 05:41:06 GMT
Organisation : Université d'Helsinki

Vous regrettez les beaux jours de Minix-1.1, quand les hommes étaient des hommes et écrivaient
leurs propres pilotes de périphériques ? Vous n'avez pas de projet intéressant et vous êtes en train de mourir
pour vous faire les dents sur un système d'exploitation que vous pouvez essayer de modifier selon vos besoins ?

Vous trouvez frustrant que tout fonctionne sur Minix ? Fini le tout-
des nuits entières pour faire fonctionner un programme astucieux ? Alors cet article pourrait bien être
pour toi :-)

Comme je l'ai mentionné il y a un mois (?), je travaille sur une version gratuite d'un
minix-lookalike pour les ordinateurs AT-386. Il a finalement atteint le stade
où il est même utilisable (même si cela peut ne pas être le cas en fonction de ce que vous voulez),
et je suis prêt à publier les sources pour une diffusion plus large. C'est juste la version 0.02 (+1 (très petit) patch déjà), mais j'ai réussi
exécutez bash/gcc/gnu-make/gnu-sed/compress etc en dessous.

Les sources de ce projet personnel peuvent être trouvées sur nic.funet.fi (128.214.6.100) dans le répertoire /pub/OS/Linux. Le répertoire
contient un fichier README et quelques binaires pour travailler sous Linux (bash, update et gcc, que demander de plus :-). Noyau complet
la source est fournie, car aucun code minix n'a été utilisé. Les sources de la bibliothèque sont
seulement partiellement gratuit, donc ne peut pas être distribué actuellement.
Le système est capable de compiler « tel quel » et fonctionne. Héhé.
Les sources des binaires (bash et gcc) peuvent être trouvées au même endroit dans
/pub/gnu.

ALERTE ! AVERTISSEMENT ! REMARQUE ! Ces sources nécessitent toujours minix-386 pour être compilées
(et gcc-1.40, peut-être 1.37.1, je n'ai pas testé), et vous avez besoin de minix pour
configurez-le si vous souhaitez l'exécuter, ce n'est donc pas encore un système autonome
pour ceux d'entre vous qui n'ont pas de minix. J'y travaille. Vous devez également être
une sorte de hacker pour le mettre en place (?), donc pour ceux qui espèrent un
alternative à minix-386, veuillez m'ignorer. Il est actuellement destiné à pirates informatiques intéressés par les systèmes d'exploitation et les 386 avec accès à minix.

Le système a besoin d'un disque dur compatible AT (IDE convient) et d'un port EGA/VGA.
vous êtes toujours intéressé, veuillez envoyer le fichier README/RELNOTES par ftp et/ou m'envoyer un e-mail
pour plus d'informations.

Je peux (enfin, presque) vous entendre vous demander "pourquoi ?". Hurd sera

dans un an (ou deux, ou le mois prochain, qui sait), et j'ai déjà
minix. Il s'agit d'un programme pour les hackers par un hacker. J'ai aimé faire
et quelqu'un pourrait aimer le regarder et même le modifier pour
leurs propres besoins. Il est encore assez petit pour comprendre, utiliser et
modifier, et j'attends avec impatience tous les commentaires que vous pourriez avoir.

Je suis également intéressé à entendre l'avis de quiconque a écrit l'un des utilitaires/fonctions de bibliothèque pour minix. Si vos efforts sont librement
distribuable (sous copyright ou même domaine public), j'aimerais entendre de votre part, afin que je puisse les ajouter au système. J'utilise Earl Chews estdio
en ce moment (merci pour un système agréable et fonctionnel Earl), et des travaux similaires
sera le bienvenu. Votre (C) sera bien sûr laissé intact. Envoyez-moi
une ligne si vous êtes prêt à me laisser utiliser votre code.

Linus

PS. à PHIL NELSON ! Je n'arrive pas à vous joindre et je continue à recevoir "erreur de transfert - domaine inconnu de fraise" ou quelque chose comme ça.

Eh bien, cela ne ressemble pas vraiment à un système, n'est-ce pas ? Cela a fonctionné,
et certaines personnes l'ont même essayé. Il y avait plusieurs bugs désagréables (et
il n'y avait pas de pilote de disquette, pas de VM, rien), et 0,02 n'était pas vraiment
très utilisable.

0,03 a été publié peu de temps après (le délai maximum était de 2 à 3 semaines)
entre les versions, même à l'époque), et 0,03 était assez utilisable.
la version suivante a été numérotée 0.10, car les choses ont commencé à fonctionner réellement
plutôt bien. Le prochain article donne une idée de ce qui s'est passé en deux
des mois de plus...

De : torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Groupes de discussion : comp.os.minix
Objet : Re : État de LINUX ?
Résumé : Toujours en version bêta
ID du message :
Date : 19 déc. 1991 23:35:45 GMT
Organisation : Université d'Helsinki

```
Dans l'article miquels@maestro.htsa.aha.nl (Miquel van Smoorenburg) écrit :
>Bonjour *,
> Je sais que certaines personnes travaillent sur un système d'exploitation
GRATUIT pour les 386/486,
>sous le nom Linux. J'ai vérifié nic.funet.fi de temps en temps, pour voir
ce qui se passait
>ce qui se passe. Cependant, pour le moment, je n'ai pas d'accès FTP, donc
je ne
>je sais ce qui se passe en ce moment. Quelqu'un pourrait-il m'en informer
s'il vous plaît
```

?

```
>Il serait peut-être préférable de donner suite à cet article, car je pense
qu'il y a
>beaucoup de personnes potentiellement intéressées lisent ce groupe. Notez
que je ne
>j'ai vraiment *un* >= 386, mais je suis sûr qu'avec le temps j'en aurai.
```

```
Linux est toujours en version bêta (bien que disponible pour les âmes
courageuses par ftp), et
a atteint la version 0.11. Elle n'est toujours pas aussi complète que
386-minix, mais meilleur à certains égards. La « fiche d'information Linux »
devrait
sera publié ici un jour par la personne qui le tient à jour.
en attendant, je vais vous donner quelques petits conseils.
```

```
Tout d’abord les mauvaises nouvelles :
```

1. Toujours pas de SCSI : les gens travaillent dessus, mais aucune date pour le moment.

Vous avez donc besoin d'un disque d'interface AT (j'ai un rapport qui dit qu'il

```
        fonctionne sur un EISA 486 avec un disque SCSI qui émule le
        Interface AT, mais c'est plus un hasard qu'autre chose :
        ISA+AT-disk est actuellement la configuration matérielle)
```

```
 Comme vous pouvez le voir, 0.11 avait déjà un petit nombre de followers. Ce
n'était pas grand-chose,
 mais ça a marché.
```

1. toujours pas d'init/login : vous entrez dans bash en tant que root au démarrage.

```
 C'était toujours la norme dans la version suivante.
```

1. bien que j'aie une machine virtuelle quelque peu fonctionnelle (pagination sur disque), ce n'est pas le cas

prêt pour le moment. Ainsi, Linux a besoin d'au moins 4 Mo pour pouvoir exécuter le

```
        Binaires GNU (en particulier gcc). Il démarre en 2 Mo, mais vous
        impossible de compiler.
```

```
 En fait, j'ai publié une version 0.11+VM juste avant Noël -91 : je
 je n'en avais pas besoin moi-même, mais les gens essayaient de compiler le
noyau dans
 2 Mo et en échec, j'ai donc dû l'implémenter. La version 0.11+VM était
 disponible uniquement pour un petit nombre de personnes souhaitant le
tester :
 Je suis toujours surpris que cela ait fonctionné aussi bien.
```

1. minix a toujours beaucoup plus d'utilisateurs : meilleur support.

1. il n'a pas été testé pendant des années par des milliers de personnes, donc il n'y a pas

il y a probablement encore pas mal de bugs.

```
Ensuite pour les bonnes choses..
```

1. C'est gratuit (copyright par moi, mais librement distribuable sous une licence)

(droit d'auteur très indulgent)

```
 Le droit d'auteur initial était en fait beaucoup plus restrictif que le GNU
 copyleft : Je n'ai autorisé aucun changement de mains d'argent en raison de
 Linux. Cela a changé avec la version 0.12.
```

1. c'est amusant de pirater.

1. système de fichiers multithreading /real/.

1. utilise les fonctionnalités 386. Ainsi enfermé dans la famille 386/486, mais

ça rend les choses plus claires quand on n'a pas à s'occuper des autres

```
        puces.
```

1. beaucoup plus... lisez mon .plan.

```
/Je/ pense que c'est mieux que Minix, mais je suis un peu partial.
ne sera jamais le genre de système d'exploitation professionnel que Hurd
sera (dans le prochain
siècle ou plus :), mais c'est un bon outil d'apprentissage (encore plus que
minix, à mon humble avis), et c'était/est amusant de travailler dessus.
```

```
             Linus (torvalds@kruuna.helsinki.fi)
```

1. — mon .plan ——————————–

UNIX gratuit pour le 386 - à venir 4QR 91 ou 1QR 92.

```
La version actuelle de Linux est la 0.11 - elle possède la plupart des
éléments d'un noyau Unix
```

```
besoins, et sera probablement publié en version 1.0 dès qu'il sera un peu
plus
plus de tests, et nous pouvons lancer une init/login. Actuellement, vous
obtenez
vidé dans un shell en tant que root au démarrage.
```

```
Linux peut être obtenu par ftp anonyme depuis 'nic.funet.fi' (128.214.6.100)
dans le répertoire '/pub/OS/Linux'. Le même répertoire contient également
des
fichiers binaires à exécuter sous Linux. Actuellement gcc, bash, update,
uemacs,
tar, make et fileutils. Plusieurs personnes ont obtenu un système
opérationnel,
mais c'est toujours un noyau de hackers.
```

```
Linux nécessite toujours un disque compatible AT pour être utile : les gens
sont
je travaille sur un pilote SCSI, mais je ne sais pas quand il sera prêt.
```

```
Il existe maintenant quelques autres sites contenant Linux, car les gens ont
J'ai eu des difficultés à me connecter à la carte réseau. Les sites sont :
      Tupac-Amaru.Informatik.RWTH-Aachen.DE (137.226.112.31) :
              répertoire /pub/msdos/replace
      tsx-11.mit.edu (18.172.1.2) :
              répertoire /pub/linux
```

```
Il existe également une liste de diffusion « Linux-activists@niksula.hut.fi
».
Pour adhérer, envoyez une demande à « Linux-activists-request@niksula.hut.fi
».
Cela ne sert à rien de m'envoyer un mail : je n'ai aucun contact réel avec
la liste de diffusion
(à part le fait d'être dessus, bien sûr).
```

```
Envoyez-moi un mail pour plus d'informations :
```

```
              Linus (torvalds@kruuna.Helsinki.FI)
```

```
La version 0.11 comporte ces nouveautés :
```

1. chargement à la demande
2. partage de code/données entre des processus non liés
3. des pilotes de disquette bien meilleurs (ils fonctionnent en fait pour la plupart)
4. corrections de bugs
5. prise en charge de Hercules/MDA/CGA/EGA/VGA
6. la console émet également des bips (WoW! Wonder-kernel 
7. mkfs/fsck/fdisk
8. Claviers américains/allemands/français/finlandais

9. vitesses de ligne réglables pour com1/2

```
 Comme vous pouvez le voir : 0.11 était en fait autonome : j'ai écrit le
premier
 programmes mkfs/fsck/fdisk pour cela, de sorte que vous n'avez pas besoin
de minix
 il ne reste plus qu'à le configurer. De plus, les lignes série avaient été
codées en dur à 2400 bps,
 car c'était tout ce que j'avais.
```

```
Il manque encore :
- initialisation/connexion
- renommer l'appel système
- tuyaux nommés
- liens symboliques
```

```
 Eh bien, ils sont tous là maintenant : init/login n'a pas encore atteint la
version 0.12,
 et rename() a été implémenté en tant que patch quelque part entre 0.12 et
 0.95. Les liens symboliques étaient dans la version 0.95, mais les pipes
nommés n'ont pas été introduits avant
 0,96.

 Remarque : le numéro de version est passé directement de 0.12 à 0.95, car
 la suite de 0.12 devenait suffisamment complète pour mériter un numéro
 dans les 0,90
```

```
La version 0.12 sortira probablement en janvier (le 15 environ) et
comportera :
- Contrôle des tâches POSIX (par tytso)
- VM (pagination sur disque)
- Corrections mineures
```

```
 En fait, la version 0.12 est sortie le 5 janvier et contenait des
corrections majeures.
 C'était en fait un noyau très stable : il fonctionnait sur de nombreuses
nouvelles
 matériel, et il n'y avait pas besoin de correctifs pendant longtemps. 0.12
était
 également le noyau qui "l'a fait" : c'est à ce moment-là que Linux a
commencé à se propager
 beaucoup plus rapide. Les versions antérieures du noyau étaient réservées
aux hackers :
 0.12 a en fait plutôt bien fonctionné.



 Remarque : Le document suivant est une réponse de Linus Torvalds, créateur
de
 Linux, dans lequel il parle de ses expériences dans les premières étapes de
```

```
 Développement Linux
```

À : Linux-Activists@BLOOM-PICAYUNE.MIT.EDU De : torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds) Objet : Re : Écrire un OS - questions !! Date : 5 mai 1992 07:58:17 GMT

Dans l'article nani@td2cad.intel.com (V. Narayanan) écrit :

Salut tout le monde,

```
       Depuis un certain temps, ce « novice » se demande comment on peut y
parvenir.
```

m à propos de la tâche d'écriture d'un système d'exploitation à partir de « zéro ». Voici donc quelques questions, et j'apprécierais si vous pouviez prendre le temps d'y répondre.

Eh bien, je vois que quelqu'un d'autre a déjà répondu, mais j'ai pensé que je prendrais le relais parties spécifiques à Linux. Ce ne sont que mes expériences personnelles, et je ne sais pas à quel point c'est normal.

1) Comment débogueriez-vous généralement le noyau pendant la phase de développement ?

Cela dépend à la fois de la machine et de l'état d'avancement du noyau : sur des systèmes plus simples, il est généralement plus facile à configurer. Voici ce que je a dû le faire sur un 386 en mode protégé.

Le pire, c'est au début : après avoir même un système minimal, vous peut utiliser printf etc, mais passer en mode protégé sur un 386 n'est pas amusant, surtout si au début vous ne connaissez pas très bien l'architecture. C'est il est terriblement facile de redémarrer le système à ce stade : si le 386 remarque que quelque chose ne va pas, il s'éteint et redémarre - vous n'avez même pas avoir une chance de voir ce qui ne va pas.

Printf() n'est pas très utile - un redémarrage efface également l'écran, et de toute façon, vous devez avoir accès à la vidéo-mémoire, ce qui pourrait échouer si votre les segments sont incorrects, etc. Ne pensez même pas aux débogueurs : non débogueur que je connais peut suivre un 386 en mode protégé. Un émulateur 386 peut faire le travail, ou du matériel lourd, mais ce n'est généralement pas le cas possible.

Ce que j'ai utilisé était une simple boucle de mise à mort : j'ai mis des déclarations comme

mourir:

```
      matrice jmp
```

à des endroits stratégiques. S'il se bloquait, vous étiez en sécurité, s'il redémarrait, vous savait au moins que cela s'était produit avant la boucle de dé. Alternativement, vous pourriez utiliser les ports IO sonores pour certains indices sonores, mais comme je n'avais aucune expérience avec le matériel PC, je ne l'ai même pas utilisé. Je ne dis pas que c'est le seule façon : je n'ai pas commencé à écrire un noyau, je voulais juste explorer les 386 primitives de commutation de tâches, etc., et c'est ainsi que j'ai commencé (en vers avril-91).

Une fois que vous avez un système minimal en place et que vous pouvez utiliser l'écran pour la sortie, il ça devient un peu plus facile, mais c'est à ce moment-là qu'il faut activer les interruptions. Bang, redémarrage instantané et retour à l'ancienne méthode. Au total, il a fallu environ 2 mois pour que je puisse bien régler les 386 choses afin que je il n'était plus nécessaire de compter sur le fait d'éviter de redémarrer immédiatement et d'avoir le éléments de base mis en place (pagination, interruption de la minuterie et un simple commutateur de tâches) pour tester les segments, etc.).

2) Pouvez-vous tester la fonctionnalité du noyau en l'exécutant en tant que processus sur un

```
  un système d'exploitation différent ? Le système d'exploitation
(l'environnement de développement) ne générerait-il pas
  exceptions dans les cas où le noyau (du nouveau système d'exploitation)
tente de modifier
  registres « privilégiés » ?
```

Oui, c'est généralement possible pour certaines choses, mais par exemple les pilotes de périphériques doivent généralement être testés sur la machine nue. J'ai utilisé minix pour je développe Linux, donc je n'avais pas accès aux registres d'E/S, aux interruptions, etc. Sous DOS, il aurait été possible d'avoir accès à tout cela, mais alors vous n'avez pas le mode 32 bits. Intel n'est pas si génial - il le serait cela aurait probablement été beaucoup plus facile sur un 68040 ou similaire.

Donc, après avoir obtenu un simple sélecteur de tâches (il basculait entre deux processus qui ont imprimé AAAA... et BBBB... respectivement en utilisant le interruption du minuteur - Mon Dieu, j'étais fier de ça), je devais quand même continuer débogage essentiellement en utilisant printf. La première chose écrite était la pilote de clavier : c'est la raison pour laquelle il est toujours entièrement écrit en assembleur (je n'ai pas encore osé passer au C - j'étais encore en train de déboguer à à propos du niveau d’instruction).

Après cela, j'ai écrit les pilotes série, et voilà, j'avais un simple programme de terminal en cours d'exécution (enfin, pas si simple en fait). C'était toujours les deux mêmes processus (AAA..), mais maintenant ils lisent et écrivent sur le console/lignes série à la place. J'ai dû redémarrer pour m'en sortir, mais c'était un simple noyau.

Après cela, c'était simple : le codage était toujours compliqué, mais j'avais quelques appareils et le débogage était plus facile. J'ai commencé à utiliser C à ce stade et cela accélère certainement le développement. C'est aussi à ce moment-là que je commence à comprendre sérieux au sujet de mes idées mégalomanes pour faire “un meilleur minix que minix”. J'espérais pouvoir recompiler gcc sous Linux un jour...

Le pilote du disque dur était plus ou moins le même : cette fois, les problèmes avec une mauvaise documentation a commencé à apparaître. Le PC est peut-être le plus utilisé l'architecture dans le monde en ce moment, mais cela ne signifie pas que les documents sont mieux : en fait, je n'ai vu aucun livre mentionnant même l'étrange Accouplement 386-387 dans un AT etc (Merci Bruce).

Après cela, un petit système de fichiers, et voilà, vous avez un Unix minimal. Deux mois pour les configurations de base, mais ensuite seulement un peu plus longtemps jusqu'à ce que j'aie un pilote de disque (sérieusement buggé, mais il fonctionnait sur ma machine) et un petit système de fichiers. C'est à peu près à ce moment-là que j'ai rendu la version 0.01 disponible (tard août-91 ? Quelque chose comme ça) : ce n'était pas joli, il n'y avait pas de disquette pilote, et il ne pouvait pas faire grand-chose. Je ne pense pas que quiconque ait jamais compilé cette version. Mais à ce moment-là, j'étais accro et je ne voulais plus Arrête jusqu'à ce que je puisse jeter Minix.

3) De nouveaux linkers et chargeurs devraient-ils être écrits avant d'obtenir une version de base ?

```
  noyau en cours d'exécution ?
```

Toutes les versions jusqu'à environ 0.11 ont été compilées de manière croisée sous minix386 - comme étaient les programmes utilisateur. J'ai finalement réussi à faire fonctionner bash et gcc sous 0,02, et bien qu'une condition de course dans le code du cache tampon m'en ait empêché en recompilant gcc avec lui-même, j'ai pu m'attaquer à des compilations plus petites. 0.03 (octobre ?) a pu recompiler gcc sous lui-même, et je pense c'est la première version que quelqu'un d'autre a réellement utilisée. Toujours pas disquettes, mais la plupart des choses de base fonctionnaient.

Après 0,03, j'ai décidé que la version suivante était réellement utilisable (elle l'était, en quelque sorte, mais bon sang, X sous 0,96 est plus impressionnant), et j'ai appelé le suivant version 0.10 (novembre ?). Il y avait encore un bug assez sérieux dans le code de gestion du cache tampon, mais après avoir corrigé cela, c'était plutôt correct. 0.11 (décembre) avait le premier pilote de disquette et était le point où j'ai J'ai commencé à faire du développement Linux sous mon propre nom. Tout aussi bien, comme je j'ai détruit ma partition minix386 par erreur en essayant de composer automatiquement /dev/hd2.

À cette époque, d'autres utilisaient déjà Linux et manquaient de ressources. mémoire. Ce qui était particulièrement triste, c'était que gcc ne fonctionnait pas sur un disque de 2 Mo machine, et bien que c386 ait été porté, il ne faisait pas tout ce que gcc faisait, et je n'ai pas pu recompiler le noyau. J'ai donc dû implémenter la pagination du disque : 0.12 est sorti en janvier (?) et j'avais la pagination ainsi que le contrôle des tâches par tytso (et d'autres patchs : pmacdona avait commencé sur VC, etc.). C'était la première version qui a commencé à avoir des fonctionnalités « non essentielles », et étant en partie écrit par d'autres. C'était aussi la première version qui en fait, il a fait beaucoup de choses mieux que Minix, et à présent, les gens ont commencé à vraiment intéressé.

Puis ce fut la version 0.95 en mars, des corrections de bugs en avril, et bientôt la version 0.96. c'était certainement amusant (et j'espère que cela continuera à l'être) - les réactions ont ont été pour la plupart très positifs, et on apprend beaucoup en faisant ce genre de choses. chose (par contre, tes études en souffrent à d'autres égards :)

```
                Linus
```

TEXTE ORIGINAL

LINUX's History

```
 Note: The following text was written by Linus on July 31 1992. It is a
 collection of various artifacts from the period in which Linux first
 began to take shape.

 This is just a sentimental journey into some of the first posts
 concerning linux, so you can happily press 'n' now if you actually
 thought you'd get anything technical.
```

```
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
```

```
Subject: Gcc-1.40 and a posix-question
Message-ID:
Date: 3 Jul 91 10:00:50 GMT
```

```
Hello netlanders,
```

```
Due to a project I'm working on (in minix), I'm interested in the posix
standard definition. Could somebody please point me to a (preferably)
machine-readable format of the latest posix rules? Ftp-sites would be
nice.
```

```
 The project was obviously linux, so by July 3rd I had started to think
 about actual user-level things: some of the device drivers were ready,
 and the harddisk actually worked. Not too much else.
```

```
As an aside for all using gcc on minix - [ deleted ]
```

```
 Just a success-report on porting gcc-1.40 to minix using the 1.37
 version made by Alan W Black & co.
```

```
              Linus Torvalds          torvalds@kruuna.helsinki.fi
```

```
PS. Could someone please try to finger me from overseas, as I've
installed a "changing .plan" (made by your's truly), and I'm not certain
it works from outside? It should report a new .plan every time.
```

```
 So I was clueless - had just learned about named pipes. Sue me. This
 part of the post got a lot more response than the actual POSIX query,
 but the query did lure out arl from the woodwork, and we mailed around
 for a bit, resulting in the Linux subdirectory on nic.funet.fi.

 Then, almost two months later, I actually had something working: I
 made sources for version 0.01 available on nic sometimes around this
 time. 0.01 sources weren't actually runnable: they were just a token
 gesture to arl who had probably started to despair about ever getting
 anything. This next post must have been from just a couple of weeks
 before that release.
```

```
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system
Message-ID:
Date: 25 Aug 91 20:57:08 GMT
Organization: University of Helsinki
```

```
Hello everybody out there using minix -
```

```
I'm doing a (free) operating system (just a hobby, won't be big and
professional like gnu) for 386(486) AT clones.  This has been brewing
```

```
since april, and is starting to get ready.  I'd like any feedback on
things people like/dislike in minix, as my OS resembles it somewhat
(same physical layout of the file-system (due to practical reasons)
among other things).
```

```
I've currently ported bash(1.08) and gcc(1.40), and things seem to work.
This implies that I'll get something practical within a few months, and
I'd like to know what features most people would want.  Any suggestions
are welcome, but I won't promise I'll implement them :-)
```

```
              Linus (torvalds@kruuna.helsinki.fi)
```

```
PS.  Yes - it's free of any minix code, and it has a multi-threaded fs.
It is NOT protable (uses 386 task switching etc), and it probably never
will support anything other than AT-harddisks, as that's all I have :-(.
```

```
 Judging from the post, 0.01 wasn't actually out yet, but it's close.
 I'd guess the first version went out in the middle of September -91. I
 got some responses to this (most by mail, which I haven't saved), and
 I even got a few mails asking to be beta-testers for linux. After that
 just a few general answers to quesions on the net:
```

```
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: Re: What would you like to see most in minix?
Summary: yes - it's nonportable
Message-ID:
Date: 26 Aug 91 11:06:02 GMT
Organization: University of Helsinki
```

```
In article  jkp@cs.HUT.FI (Jyrki Kuoppala) writes:
>> [re: my post about my new OS]
>
>Tell us more!  Does it need a MMU?
```

```
Yes, it needs a MMU (sorry everybody), and it specifically needs a
386/486 MMU (see later).
```

```
>
>>PS.  Yes - it's free of any minix code, and it has a multi-threaded fs.
>>It is NOT protable (uses 386 task switching etc)
>
>How much of it is in C?  What difficulties will there be in porting?
>Nobody will believe you about non-portability ;-), and I for one would
>like to port it to my Amiga (Mach needs a MMU and Minix is not free).
```

```
Simply, I'd say that porting is impossible.  It's mostly in C, but most
people wouldn't call what I write C.  It uses every conceivable feature
```

of the 386 I could find, as it was also a project to teach me about the 386. As already mentioned, it uses a MMU, for both paging (not to disk yet) and segmentation. It's the segmentation that makes it REALLY 386 dependent (every task has a 64Mb segment for code & data - max 64 tasks in 4Gb. Anybody who needs more than 64Mb/task - tough cookies).

It also uses every feature of gcc I could find, specifically the __asm__ directive, so that I wouldn't need so much assembly language objects. Some of my "C"-files (specifically mm.c) are almost as much assembler as C. It would be "interesting" even to port it to another compiler (though why anybody would want to use anything other than gcc is a mystery).

Note: linux has in fact gotten more portable with newer versions: there was a lot more assembly in the early versions. It has in fact been ported to other architectures by now.

Unlike minix, I also happen to LIKE interrupts, so interrupts are handled without trying to hide the reason behind them (I especially like my hard-disk-driver. Anybody else make interrupts drive a state-machine?). All in all it's a porters nightmare.

>As for the features; well, pseudo ttys, BSD sockets, user-mode
>filesystems (so I can say cat /dev/tcp/kruuna.helsinki.fi/finger),
>window size in the tty structure, system calls capable of supporting
>POSIX.1. Oh, and bsd-style long file names.

Most of these seem possible (the tty structure already has stubs for window size), except maybe for the user-mode filesystems. As to POSIX, I'd be delighted to have it, but posix wants money for their papers, so that's not currently an option. In any case these are things that won't be supported for some time yet (first I'll make it a simple minix-lookalike, keyword SIMPLE).

Linus (torvalds@kruuna.helsinki.fi)

PS. To make things really clear - yes I can run gcc on it, and bash, and most of the gnu [bin/file]utilities, but it's not very debugged, and the library is really minimal. It doesn't even support floppy-disks yet. It won't be ready for distribution for a couple of months. Even then it probably won't be able to do much more than minix, and much less in some respects. It will be free though (probably under gnu-license or similar).

Well, obviously something worked on my machine: I doubt I had yet gotten gcc to compile itself under linux (or I would have been too proud of it not to mention it). Still before any release-date.

Then, October 5th, I seem to have released 0.02. As I already mentioned, 0.01 didn't actually come with any binaries: it was just source code for people interested in what linux looked like. Note the lack of announcement for 0.01: I wasn't too proud of it, so I think I

only sent a note to everybody who had shown interest.

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: Free minix-like kernel sources for 386-AT
Message-ID:
Date: 5 Oct 91 05:41:06 GMT
Organization: University of Helsinki

Do you pine for the nice days of minix-1.1, when men were men and wrote their own device drivers? Are you without a nice project and just dying to cut your teeth on a OS you can try to modify for your needs? Are you finding it frustrating when everything works on minix? No more all-nighters to get a nifty program working? Then this post might be just for you :-)

As I mentioned a month(?) ago, I'm working on a free version of a minix-lookalike for AT-386 computers. It has finally reached the stage where it's even usable (though may not be depending on what you want), and I am willing to put out the sources for wider distribution. It is just version 0.02 (+1 (very small) patch already), but I've successfully run bash/gcc/gnu-make/gnu-sed/compress etc under it.

Sources for this pet project of mine can be found at nic.funet.fi (128.214.6.100) in the directory /pub/OS/Linux. The directory also contains some README-file and a couple of binaries to work under linux (bash, update and gcc, what more can you ask for :-). Full kernel source is provided, as no minix code has been used. Library sources are only partially free, so that cannot be distributed currently. The system is able to compile "as-is" and has been known to work. Heh. Sources to the binaries (bash and gcc) can be found at the same place in /pub/gnu.

ALERT! WARNING! NOTE! These sources still need minix-386 to be compiled (and gcc-1.40, possibly 1.37.1, haven't tested), and you need minix to set it up if you want to run it, so it is not yet a standalone system for those of you without minix. I'm working on it. You also need to be something of a hacker to set it up (?), so for those hoping for an alternative to minix-386, please ignore me. It is currently meant for hackers interested in operating systems and 386's with access to minix.

The system needs an AT-compatible harddisk (IDE is fine) and EGA/VGA. If you are still interested, please ftp the README/RELNOTES, and/or mail me for additional info.

I can (well, almost) hear you asking yourselves "why?". Hurd will be out in a year (or two, or next month, who knows), and I've already got minix. This is a program for hackers by a hacker. I've enjouyed doing it, and somebody might enjoy looking at it and even modifying it for

```
their own needs.  It is still small enough to understand, use and
modify, and I'm looking forward to any comments you might have.
```

```
I'm also interested in hearing from anybody who has written any of the
utilities/library functions for minix. If your efforts are freely
distributable (under copyright or even public domain), I'd like to hear
from you, so I can add them to the system. I'm using Earl Chews estdio
right now (thanks for a nice and working system Earl), and similar works
will be very wellcome. Your (C)'s will of course be left intact. Drop me
a line if you are willing to let me use your code.
```

```
               Linus
```

```
PS. to PHIL NELSON! I'm unable to get through to you, and keep getting
"forward error - strawberry unknown domain" or something.
```

```
 Well, it doesn't sound like much of a system, does it? It did work,
 and some people even tried it out. There were several bad bugs (and
 there was no floppy-driver, no VM, no nothing), and 0.02 wasn't really
 very useable.

 0.03 got released shortly thereafter (max 2-3 weeks was the time
 between releases even back then), and 0.03 was pretty useable. The
 next version was numbered 0.10, as things actually started to work
 pretty well. The next post gives some idea of what had happened in two
 months more...
```

```
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: Re: Status of LINUX?
Summary: Still in beta
Message-ID:
Date: 19 Dec 91 23:35:45 GMT
Organization: University of Helsinki
```

```
In article  miquels@maestro.htsa.aha.nl (Miquel van Smoorenburg) writes:
>Hello *,
>     I know some people are working on a FREE O/S for the 386/486,
>under the name Linux. I checked nic.funet.fi now and then, to see what was
>happening. However, for the time being I am without FTP access so I don't
>know what is going on at the moment. Could someone please inform me about
it
```

?

```
>It's maybe best to follow up to this article, as I think that there are
>a lot of potential interested people reading this group. Note, that I don't
>really *have* a >= 386, but I'm sure in time I will.
```

```
Linux is still in beta (although available for brave souls by ftp), and
```

```
has reached the version 0.11.  It's still not as comprehensive as
386-minix, but better in some respects.  The "Linux info-sheet" should
be posted here some day by the person that keeps that up to date.  In
the meantime, I'll give some small pointers.
```

```
First the bad news:
```

1. Still no SCSI: people are working on that, but no date yet.

Thus you need a AT-interface disk (I have one report that it

```
       works on an EISA 486 with a SCSI disk that emulates the
       AT-interface, but that's more of a fluke than anything else:
       ISA+AT-disk is currently the hardware setup)
```

```
As you can see, 0.11 had already a small following. It wasn't much,
but it did work.
```

1. still no init/login: you get into bash as root upon bootup.

```
That was still standard in the next release.
```

1. although I have a somewhat working VM (paging to disk), it's not

ready yet. Thus linux needs at least 4M to be able to run the

```
       GNU binaries (especially gcc).  It boots up in 2M, but you
       cannot compile.
```

```
I actually released a 0.11+VM version just before Christmas -91: I
didn't need it myself, but people were trying to compile the kernel in
2MB and failing, so I had to implement it. The 0.11+VM version was
available only to a small number of people that wanted to test it out:
I'm still surprised it worked as well as it did.
```

1. minix still has a lot more users: better support.

1. it hasn't got years of testing by thousands of people, so there

are probably quite a few bugs yet.

```
Then for the good things..
```

1. It's free (copyright by me, but freely distributable under a

very lenient copyright)

```
The early copyright was in fact much more restrictive than the GNU
copyleft: I didn't allow any money at all to change hands due to
linux. That changed with 0.12.
```

1. it's fun to hack on.

1. /real/ multithreading filesystem.

1. uses the 386-features. Thus locked into the 386/486 family, but

it makes things clearer when you don't have to cater to other

```
        chips.
```

1. a lot more... read my .plan.

```
/I/ think it's better than minix, but I'm a bit prejudiced.  It will
never be the kind of professional OS that Hurd will be (in the next
century or so :), but it's a nice learning tool (even more so than
minix, IMHO), and it was/is fun working on it.
```

```
              Linus (torvalds@kruuna.helsinki.fi)
```

1. — my .plan ——————————–

Free UNIX for the 386 - coming 4QR 91 or 1QR 92.

```
The current version of linux is 0.11 - it has most things a unix kernel
needs, and will probably be released as 1.0 as soon as it gets a little
more testing, and we can get a init/login going. Currently you get
dumped into a shell as root upon bootup.
```

```
Linux can be gotten by anonymous ftp from 'nic.funet.fi' (128.214.6.100)
in the directory '/pub/OS/Linux'.  The same directory also contains some
binary files to run under Linux.  Currently gcc, bash, update, uemacs,
tar, make and fileutils.  Several people have gotten a running system,
but it's still a hackers kernel.
```

```
Linux still requires a AT-compatible disk to be useful: people are
working on a SCSI-driver, but I don't know when it will be ready.
```

```
There are now a couple of other sites containing linux, as people have
had difficulties with connecting to nic. The sites are:
      Tupac-Amaru.Informatik.RWTH-Aachen.DE (137.226.112.31):
              directory /pub/msdos/replace
      tsx-11.mit.edu (18.172.1.2):
              directory /pub/linux
```

```
There is also a mailing list set up 'Linux-activists@niksula.hut.fi'.
To join, mail a request to 'Linux-activists-request@niksula.hut.fi'.
It's no use mailing me: I have no actual contact with the mailing-list
(other than being on it, naturally).
```

```
Mail me for more info:
```

```
              Linus (torvalds@kruuna.Helsinki.FI)
```

```
0.11 has these new things:
```

1. demand loading
2. code/data sharing between unrelated processes
3. much better floppy drivers (they actually work mostly)
4. bug-corrections
5. support for Hercules/MDA/CGA/EGA/VGA
6. the console also beeps (WoW! Wonder-kernel 
7. mkfs/fsck/fdisk
8. US/German/French/Finnish keyboards
9. settable line-speeds for com1/2

```
 As you can see: 0.11 was actually stand-alone: I wrote the first
 mkfs/fsck/fdisk programs for it, so that you didn't need minix any
 more to set it up. Also, serial lines had been hard-coded to 2400bps,
 as that was all I had.
```

```
Still lacking:
- init/login
- rename system call
- named pipes
- symbolic links
```

```
 Well, they are all there now: init/login didn't quite make it to 0.12,
 and rename() was implemented as a patch somewhere between 0.12 and
 0.95. Symlinks were in 0.95, but named pipes didn't make it until
 0.96.

 Note: The version number went directly from 0.12 to 0.95, as the
 follow-on to 0.12 was getting feature-full enough to deserve a number
 in the 0.90's
```

```
0.12 will probably be out in January (15th or so), and will have:
- POSIX job control (by tytso)
- VM (paging to disk)
- Minor corrections
```

```
 Actually, 0.12 was out January 5th, and contained major corrections.
 It was in fact a very stable kernel: it worked on a lot of new
 hardware, and there was no need for patches for a long time. 0.12 was
 also the kernel that "made it": that's when linux started to spread a
 lot faster. Earlier kernel releases were very much only for hackers:
 0.12 actually worked quite well.



 Note: The following document is a reply by Linus Torvalds, creator of
 Linux, in which he talks about his experiences in the early stages of
```

```
 Linux development
```

To: Linux-Activists@BLOOM-PICAYUNE.MIT.EDU From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds) Subject: Re: Writing an OS - questions !! Date: 5 May 92 07:58:17 GMT

In article nani@td2cad.intel.com (V. Narayanan) writes:

Hi folks,

```
       For quite some time this "novice" has been wondering as to how one
goe
```

s about the task of writing an OS from “scratch”. So here are some questions, and I would appreciate if you could take time to answer 'em.

Well, I see someone else already answered, but I thought I'd take on the linux-specific parts. Just my personal experiences, and I don't know how normal those are.

1) How would you typically debug the kernel during the development phase?

Depends on both the machine and how far you have gotten on the kernel: on more simple systems it's generally easier to set up. Here's what I had to do on a 386 in protected mode.

The worst part is starting off: after you have even a minimal system you can use printf etc, but moving to protected mode on a 386 isn't fun, especially if you at first don't know the architecture very well. It's distressingly easy to reboot the system at this stage: if the 386 notices something is wrong, it shuts down and reboots - you don't even get a chance to see what's wrong.

Printf() isn't very useful - a reboot also clears the screen, and anyway, you have to have access to video-mem, which might fail if your segments are incorrect etc. Don't even think about debuggers: no debugger I know of can follow a 386 into protected mode. A 386 emulator might do the job, or some heavy hardware, but that isn't usually feasible.

What I used was a simple killing-loop: I put in statements like

die:

```
      jmp die
```

at strategic places. If it locked up, you were ok, if it rebooted, you knew at least it happened before the die-loop. Alternatively, you might use the sound io ports for some sound-clues, but as I had no experience with PC hardware, I didn't even use that. I'm not saying this is the only way: I didn't start off to write a kernel, I just wanted to explore the 386 task-switching primitives etc, and that's how I started off (in about April-91).

After you have a minimal system up and can use the screen for output, it gets a bit easier, but that's when you have to enable interrupts. Bang, instant reboot, and back to the old way. All in all, it took about 2 months for me to get all the 386 things pretty well sorted out so that I no longer had to count on avoiding rebooting at once, and having the basic things set up (paging, timer-interrupt and a simple task-switcher to test out the segments etc).

2) Can you test the kernel functionality by running it as a process on a

```
different OS?  Wouldn't the OS(the development environment) generate
exceptions in cases when the kernel (of the new OS) tries to modify
'priviledged' registers?
```

Yes, it's generally possible for some things, but eg device drivers usually have to be tested out on the bare machine. I used minix to develop linux, so I had no access to IO registers, interrupts etc. Under DOS it would have been possible to get access to all these, but then you don't have 32-bit mode. Intel isn't that great - it would probably have been much easier on a 68040 or similar.

So after getting a simple task-switcher (it switched between two processes that printed AAAA... and BBBB... respectively by using the timer-interrupt - Gods I was proud over that), I still had to continue debugging basically by using printf. The first thing written was the keyboard driver: that's the reason it's still written completely in assembler (I didn't dare move to C yet - I was still debugging at about instruction-level).

After that I wrote the serial drivers, and voila, I had a simple terminal program running (well, not that simple actually). It was still the same two processes (AAA..), but now they read and wrote to the console/serial lines instead. I had to reboot to get out of it all, but it was a simple kernel.

After that is was plain sailing: hairy coding still, but I had some devices, and debugging was easier. I started using C at this stage, and it certainly speeds up developement. This is also when I start to get serious about my megalomaniac ideas to make “a better minix that minix”. I was hoping I'd be able to recompile gcc under linux some day...

The harddisk driver was more of the same: this time the problems with bad documentation started to crop up. The PC may be the most used architecture in the world right now, but that doesn't mean the docs are any better: in fact I haven't seen /any/ book even mentioning the weird 386-387 coupling in an AT etc (Thanks Bruce).

After that, a small filesystem, and voila, you have a minimal unix. Two months for basic setups, but then only slightly longer until I had a disk-driver (seriously buggy, but it happened to work on my machine) and a small filesystem. That was about when I made 0.01 available (late august-91? Something like that): it wasn't pretty, it had no floppy driver, and it couldn't do much anything. I don't think anybody ever compiled that version. But by then I was hooked, and didn't want to stop until I could chuck out minix.

3) Would new linkers and loaders have to be written before you get a basic

```
kernel running?
```

All versions up to about 0.11 were crosscompiled under minix386 - as were the user programs. I got bash and gcc eventually working under 0.02, and while a race-condition in the buffer-cache code prevented me from recompiling gcc with itself, I was able to tackle smaller compiles. 0.03 (October?) was able to recompile gcc under itself, and I think that's the first version that anybody else actually used. Still no floppies, but most of the basic things worked.

Afetr 0.03 I decided that the next version was actually useable (it was, kind of, but boy is X under 0.96 more impressive), and I called the next version 0.10 (November?). It still had a rather serious bug in the buffer-cache handling code, but after patching that, it was pretty ok. 0.11 (December) had the first floppy driver, and was the point where I started doing linux developement under itself. Quite as

well, as I trashed my minix386 partition by mistake when trying to autodial /dev/hd2.

By that time others were actually using linux, and running out of memory. Especially sad was the fact that gcc wouldn't work on a 2MB machine, and although c386 was ported, it didn't do everything gcc did, and couldn't recompile the kernel. So I had to implement disk-paging: 0.12 came out in January (?) and had paging by me as well as job control by tytso (and other patches: pmacdona had started on VC's etc). It was the first release that started to have “non-essential” features, and being partly written by others. It was also the first release that actually did many things better than minix, and by now people started to really get interested.

Then it was 0.95 in March, bugfixes in April, and soon 0.96. It's certainly been fun (and I trust will continue to be so) - reactions have been mostly very positive, and you do learn a lot doing this type of thing (on the other hand, your studies suffer in other respects :)

```
                Linus
```