

ScreenClock

Thomas Igracki

COLLABORATORS

	TITLE : ScreenClock		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY	Thomas Igracki	March 26, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	ScreenClock	1
1.1	ScreenClock Anleitung	1
1.2	ScreenClock/Inhalt des Archivs	1
1.3	ScreenClock/Was macht das Programm?	1
1.4	ScreenClock/Aufruf des Programms	2
1.5	ScreenClock/Argument	3
1.6	ScreenClock/Argument	3
1.7	ScreenClock/Argument	4
1.8	ScreenClock/Argument	5
1.9	ScreenClock/Argument	5
1.10	ScreenClock/Argument	5
1.11	ScreenClock/Argument	6
1.12	ScreenClock/Argument	6
1.13	ScreenClock/Argument	6
1.14	ScreenClock/Argument	6
1.15	ScreenClock/Argument	7
1.16	ScreenClock/Argument	7
1.17	ScreenClock/Argument	8
1.18	ScreenClock/Argument	8
1.19	ScreenClock/Fehlermeldungen	8
1.20	ScreenClock/Programmentwicklung	9
1.21	ScreenClock/Wie kann man mich erreichen?	12
1.22	SysInfo.library by Petri Nordlund	13

Chapter 1

ScreenClock

1.1 ScreenClock Anleitung

```
ScreenClock v1.83 / von Thomas Igracki
-----/
AFD - C O P Y R I G H T

Inhalt des Archivs      Programmentwicklung
Was macht das Programm? Wie kann man mich erreichen?
Aufruf des Programms    Mögliche Fehlermeldungen
```

Diese Software unterliegt der "Standard Amiga FD-Software Copyright Note"
Sie ist GIFTWARE wie definiert in Absatz 4g. [...]
Für mehr Informationen lies bitte "AFD-COPYRIGHT" (Version 1 oder höher).

1.2 ScreenClock/Inhalt des Archivs

Folgende Dateien sollten sich in diesem Archiv befinden:

env/ScreenClock	--> Beispiel Environment-Variable
libs/sysinfo.library	--> SysInfo.library von Petri Nordlund
help/deutsch/ScreenClock.guide	--> deutsche Dokumentation, AmigaGuide Format
help/deutsch/AFD-COPYRIGHT.de	--> deutscher COPYRIGHT-Text
help/english/ScreenClock.guide	--> englische Dokumentation, AmigaGuide Format
help/english/AFD-COPYRIGHT	--> englischer COPYRIGHT-Text
Product-Info	--> Produkt-Info
ScreenClock.readme	--> kurze englische Doku, ASCII
ScreenClock.liesmich	--> kurze deutsche Doku, ASCII
ScreenClock.info	--> das WB-Icon für ScreenClock
ScreenClock	--> das Programm für OS >= v38 (2.1)

1.3 ScreenClock/Was macht das Programm?

Dieses Programm zeigt in der Titelleiste (links neben dem TiefenSymbol) des jeweils aktuellen Bildschirms das Datum und die Uhrzeit in der aktuellen Sprache an! Sowie auf Wunsch den freien Speicher und die aktuelle oder letzte (letzten 60 Sekunden) Belastung deiner CPU!

Die Features sind:

- konfigurierbares Datums- und Zeit-Format
- optionales anzeigen des freien Speichers
- ein-/ausschliessen von Bildschirmen die benutzt bzw. nicht benutzt werden sollen
- einstellbare Update-Rate in der die Anzeige erneuert werden soll
- Benutzung einer "überwachten" Umgebungs-Varaiablen (SCREENCLOCK), die beim Start und bei Verändern eingelesen wird
- Überwachung von ENV:SYS/locale.prefs, um mitzukriegen, wann die Sprache verändert wird
- optional werden nur Public-Screens benutzt, siehe OPS=OnlyPubScreens/T
- einstellbare Task Priorität, siehe Pri=TaskPriority/K/N
- AutoScroll Screens werden unterstützt, siehe AV=AlwaysVisible/T
- vl.72: 1/50 Sekunden werden auf Wunsch angezeigt, siehe ST=ShowTicks/T
- vl.73: Verdoppeln der 1/50 Sekunden auf Wunsch (für pseudo 1/100 Sek.), siehe DT=DoubleTicks/T
- vl.8 : optionales Anzeigen der aktuellen CPU Belastung, CPU=CpuUsage/T unterdrücken des "locken" des Screens während des Schreibens, siehe NLS=NoLockScreen/T
- vl.81: nur einen Fehler beseitigt (der pubScreen wurde nicht unlocked, wenn NLS=NoLockScreen/T aus ist)
- vl.82: Fix: Mit dem neuen Interface für die SysInfo.library compiliert, da sich die Schreibweise der Library geändert hat (sysinfo -> SysInfo).
Fix: Ich vergaß zu überprüfen, ob überhaupt CpuUsageLastsec implementiert ist. Wenn es nicht implementiert ist, ist kein CPU-Anzeige möglich.
- vl.83: Neu: ScreenClock fragt nun nicht mehr das ShowTitleFlag eines Screens ab, so daß ScreenClock auch auf MUI-Screens seinen Dienst verrichtet.
Neu: Man kann jetzt zwischen der CPU Belastung der letzten 60 Sekunden (recent) oder der letzten Sekunde wählen RecentCPU/T

Bei nochmaligen Starten oder break'en mit CtrlC wird ScreenClock beendet.

1.4 ScreenClock/Aufruf des Programms

Start von einer Shell:

```
[Run] ScreenClock DF=DateFormat/K,MF=MemoryFormat/K,
Screens/K,D=Delay/K/N,Mem=Memory/T,
OPS=OnlyPubScreens/T,Pri=TaskPriority/K/N,
AV=AlwaysVisible/T,ST=ShowTicks/T,DT=DoubleTicks/T
CPU=CpuUsage/T,CF=CpuFormat/K,RecentCPU/T
NLS=NoLockScreen/T
```

Start von der Workbench:

Einfach das Piktogramm doppelklicken.

Zur Zeit werden keine ToolTypes unterstützt, man sollte stattdessen die Umgebungsvariable benutzen!

1.5 ScreenClock/Argument

DF=DateFormat/K

Hier kann das Format für die Anzeige des Datums angegeben werden.

Dabei sind folgende Platzhalter verfügbar (aus `Locale.FormatDate()`):

```
%a - abgekürzter Wochentag
%A - Wochentag
%b - abgekürzter Monatsname
%B - Monatsname
%c - das gleiche wie "%a %b %d %H:%M:%S %Y"
%C - das gleiche wie "%a %b %e %T %Z %Y"
%d - Tageszahl mit führenden 0s
%D - das gleiche wie "%m/%d/%y"
%e - Tageszahl mit führenden Leerzeichen
%h - abgekürzter Monatsname
%H - Stunde mit 24-Stundenanzeige und führenden 0s
%I - Stunde mit 12-Stundenanzeige und führenden 0s
%j - Julian Datum
%m - Monatszahl mit führenden 0s
%M - Anzahl von Minuten mit führenden 0s
%n - Ein Linefeed
%p - AM oder PM Strings
%q - Stunde mit 24-Stundenanzeige
%Q - Stunde mit 12-Stundenanzeige
%r - das gleiche wie "%I:%M:%S %p"
%R - das gleiche wie "%H:%M"
%S - Anzahl von Sekunden mit führenden 0s
%t - Fügt ein TAB ein
%T - das gleiche wie "%H:%M:%S"
%U - Wochenzahl, Sonntag = erster Tag der Woche
%w - Wochentagszahl
%W - Wochenzahl, Montag = erster Tag der Woche
%x - das gleiche wie "%m/%d/%y"
%X - das gleiche wie "%H:%M:%S"
%y - Jahr mit 2 Zeichen und führenden 0s
%Y - Jahr mit 4 Zeichen und führenden 0s
```

Voreingestellt ist: "%A %d-%h-%y %T".

Das entspricht dieser Anzeige: Montag 09-Jan-95 18:22:57.

Wenn man während ScreenClock läuft, diesen Parameter auf "" setzt, wird wieder der voreingestellte String benutzt!

1.6 ScreenClock/Argument

MF=MemoryFormat/K

Hier kann das Format für die Anzeige des freien Speichers angegeben werden.

Es werden 3 Platzhalter (%ld bzw. %lD) unterstützt:

- der 1. steht für den freien CHIP-Speicher in KB,
- der 2. steht für den freien FAST-Speicher in KB und
- der 3. steht für den gesamten freien Speicher in KB.

Voreingestellt ist: " - C:%5lD F:%5lD".

Das entspricht folgender Speicher-Anzeige: " - C:1.737 F:3.538".

Wenn man während ScreenClock läuft, diesen Parameter auf "" setzt, wird wieder der voreingestellte String benutzt!

Man sollte darauf achten, daß das 1. Zeichen ein Leerzeichen o.ä. ist, da der erzeugte String direkt hinter dem Datum steht!

Man kann natürlich auch an dem Datum ein Leerzeichen anfügen;-)

Tip:

Wenn man die Reihenfolge ändern möchte, muß man zwischen dem % und dem ld bzw. lD Zeichen, ein <Position>\$ benutzen!
<Position> gibt den Platzhalter an.

Beispiel: MF = " Total: %3\$lD (FAST = %2\$lD, CHIP = %1\$lD)"

Wenn man z.B. nur den 3. Platzhalter benutzen will, muß man folgende Schreibweise benutzen: MF = "%0.0ld%0.0ld Total: %lD"!

Siehe dazu auch Locale/FormatString().

Der Unterschied zwischen %ld und %lD ist, daß beim zweiten das lokale Format benutzt wird.

1.7 ScreenClock/Argument

Screens/K

Gibt ein Muster (mit AmigaDOS Wildcards) für den Bildschirmtitel an. ScreenClock erscheint dann nur auf diesen Bildschirmen!

Da mit dem 'defaultTitle' des Bildschirms verglichen wird, funktioniert die Uhr auch wenn ein anderes Programm seinen Fenstertitel im Screen anzeigt!

Wird kein Muster angegeben, werden alle Bildschirme benutzt, die einen Titel besitzen (MuchMore z.B. nicht!).

Wenn man während ScreenClock läuft, diesen Parameter auf "" setzt, wird dieser Vorgang abgeschaltet!

Der Bildschirm kann während ScreenClock läuft problemlos geschlossen werden! Die Daten werden nämlich direkt in den RastPort der Titelleiste geschrieben, es wird also kein Fenster geöffnet!

Das hat den Vorteil, daß so auch Bildschirme geschlossen werden können,

bei denen das Programm vor dem Schließen des Bildschirm prüft, ob noch Fenster drauf sind (z.B. CygnusEd, Term)!

Tip1:

Man sollte anstatt dem Stern * lieber ein #? benutzen, da der Stern in Verbindung mit einem " noch andere Bedeutungen haben kann!

Tip2:

Wenn bestimmte Screens nicht benutzt werden sollen, muß man foldenden Pattern benutzen: ~(<TitelName>|<TitelName>)

Beispiel: SCREENS = "~(DosControl#?)"

Wirkung : Der 'DosControl'-Screen wird nicht benutzt.

1.8 ScreenClock/Argument

D=Delay/N

Diese Zahl gibt an, in welchem Zeitabstand ScreenClock die Anzeige erneuert.

Anzahl von 1/50 Sekunden. Mindestens 10, maximal 100.

Voreingestellt ist 50.

1.9 ScreenClock/Argument

Mem=Memory/T

Hier gibt man an, ob zusätzlich der freie Speicher (MemoryFormat) angezeigt werden soll (immer hinter dem Datum!).

Anzeige an : Memory = ON oder Mem = ON

Anzeige aus: Memory = OFF oder Mem = OFF

Voreingestellt ist OFF.

Das "=" kann auch weggelassen werden.

1.10 ScreenClock/Argument

OPS=OnlyPubScreens/T

Hier gibt man an, ob nur (nicht private!) Public-Screens benutzt werden sollen.

AN : OnlyPubScreens = ON oder OPS = ON

AUS: OnlyPubScreens = OFF oder OPS = OFF

Voreingestellt ist OFF.

Das "=" kann auch weggelassen werden.

1.11 ScreenClock/Argument

Pri=TaskPriority/K/N

Hier gibt man die Priorität von ScreenClock an.

Es werden nur Prioritäten zwischen -3 und +3 akzeptiert.

Voreingestellt ist 0.

1.12 ScreenClock/Argument

AV=AlwaysVisible/T

Hiermit gibt man an, ob die Anzeige im sichtbaren Teil des Bildschirms (ON) oder immer links neben dem TiefenSymbol (OFF).

AlwaysVisible = ON oder AV ON -> immer im sichtbaren Bereich!

AlwaysVisible = OFF oder AV OFF -> immer links vom TiefenSymbol

Voreingestellt ist ON!

Das "=" kann auch weggelassen werden.

1.13 ScreenClock/Argument

ST=ShowTicks/T

Hiermit gibt man an, ob 1/50 Sekunden angezeigt werden sollen.

ShowTicks = ON oder ST ON -> ja, sie werden gezeigt

ShowTicks = OFF oder ST OFF -> nein, sie werden nicht gezeigt

Voreingestellt ist OFF!

Das "=" kann auch weggelassen werden.

Hinweis:

Die 1/50 Sekunden werden getrennt von einen "." an das DatumFormat angehängt.

Das wurde so gelöst, da es nicht von Locale.FormatDate() unterstützt wird, und ich denke, daß es nicht so viele Leute benutzen werden, so daß sich der Aufwand nicht lohnen würde es überall im String zu erlauben (per neuem Platzhalter und erweitern von Locale.FormatDate()).

Falls es doch mehrere benutzen wollen und sie ihre Uhrzeit nicht ganz rechts in ihrem Datum haben, könnte ich mich auch überreden lassen;-)

1.14 ScreenClock/Argument

DT=DoubleTicks/T

Hiermit gibt man an, ob die 1/50 Sekunden mit 2 multipliziert werden sollen.
So daß man pseudo 1/100 Sekunden hat.

DoubleTicks = ON oder DT ON -> ja, sie werden verdoppelt

DoubleTicks = OFF oder DT OFF -> nein, sie werden nicht verdoppelt

Voreingestellt ist OFF!

Das "=" kann auch weggelassen werden.

Hinweis:

Diese Option arbeitet nur in Verbindung mit ShowTicks/T!

1.15 ScreenClock/Argument

CPU=CpuUsage/T

Hiermit gibt man an, ob zusätzlich die aktuelle Belastung der CPU in %
(CPU-Format) ausgegeben werden soll (immer hinter dem freien Speicher
bzw. dem Datum!).

Für die letzte (letzten 60 Sekunden) CPU-Belastung, siehe RecentCPU.

Das geschieht mit Hilfe der sysinfo.library von Petri Nordlund.

CpuUsage = ON oder CPU ON -> sie wird angezeigt

CpuUsage = OFF oder CPU OFF -> sie wird nicht angezeigt

Voreingestellt ist OFF!

Das "=" kann auch weggelassen werden.

1.16 ScreenClock/Argument

CF=CpuFormat/K

Hiermit wird das Format für die Anzeige der aktuellen CPU-Belastung
angegeben.

Für die letzte (letzten 60 Sekunden) CPU-Belastung, siehe RecentCPU.

Es werden maximal 2 Platzhalter (%ld) unterstützt:

- der 1. steht für die Zahl vor dem Punkt
- der 2. steht für die Zahl nach dem Punkt

Siehe dazu auch Locale/FormatString().

Voreingestellt ist: " - CPU = %2ld.%.1ld%".

Das entspricht z.B. folgender Anzeige: " - CPU = 2.7%".

Wenn man während ScreenClock läuft, diesen Parameter auf "" setzt, wird
wieder der voreingestellte String benutzt!

Man sollte darauf achten, daß das 1. Zeichen ein Leerzeichen o.ä. ist,

da der erzeugte String direkt hinter dem freien Speicher steht!

Tip:

Wenn man keine Nachkommastelle haben will, kann man auch nur einen Platzhalter angeben. Das sähe dann so aus: `CF = " CPU = %2ld"`

1.17 ScreenClock/Argument

RecentCPU/T

Hiermit gibt man an, ob statt der aktuellen Belastung die der letzten 60 Sek. ausgegeben werden soll (immer hinter dem freien Speicher bzw. dem Datum!).

Das geschieht mit Hilfe der `sysinfo.library` von Petri Nordlund.

RecentCPU = ON -> die letzte CPU-Belastung wird angezeigt

RecentCPU = OFF -> die aktuelle CPU-Belastung wird angezeigt

Voreingestellt ist OFF!

Das "=" kann auch weggelassen werden.

1.18 ScreenClock/Argument

NLS=NoLockScreen/T

Hiermit gibt man an, ob während des Schreibens der Daten in die Titelleiste der Screen "gelockt" werden soll.

Dadurch konnte es zu Problemen kommen, wenn nur ein Screen offen ist und der ScreenModus verändert wird (jedenfalls bei Marcin Andrzejak).

NoLockScreen = ON oder NLS ON -> es wird nicht gelockt

NoLockScreen = OFF oder NLS OFF -> es wird gelockt

Voreingestellt ist OFF!

1.19 ScreenClock/Fehlermeldungen

Mögliche Fehlermeldungen während des Startens sind:

- 1 : Der MsgPort konnte nicht erstellt werden.
- 2 : `Dos AllocDosObject(Dos.rdArgs)` schlug fehl.
- 3 : Notify auf ENV:ScreenClock schlug fehl.
- 4 : Notify auf ENV:SYS/locale.prefs schlug fehl.
- 5 : `InitSysinfo()` konnte nicht aufgerufen werden.

Falls die Umgebungsvariable oder die Shell-Argumente nicht geparsed werden konnten, kommt eine entsprechende Fehler-Meldung.

1.20 ScreenClock/Programmentwicklung

Release 1.83:

Revision 38.27 1996/05/26 19:08:33 tig

Neu: ScreenClock fragt nun nicht mehr das ShowTitleFlag eines Screens ab, so daß ScreenClock auch auf MUI-Screens seinen Dienst verrichtet.

Neu: Man kann jetzt zwischen der CPU Belastung der letzten 60 Sekunden (recent) oder der letzten Sekunde wählen

Release 1.82:

38.26 (02.12.95):

Fix: Neu compiliert mit dem Interface für die SysInfo.library, da sich die Schreibweise der Library geändert hat (sysinfo -> SysInfo).

Fix: Ich vergaß zu überprüfen, ob überhaupt CpuUsageLastsec implementiert ist. Wenn es nicht implementiert ist, ist kein CPU-Anzeige möglich.

Release 1.81:

38.25 (25.11.95):

Fix: einen Fehler beseitigt (der pubScreen wurde nicht unlocked, wenn 'NoLockScreen' aus ist)

Release 1.8:

38.24 (15.11.95):

Neu: Option 'NLS=NoLockScreen/T' gibt an, ob der Screen während des Schreibens der Daten in die Titelleiste "gelockt" werden soll. Dadurch konnte es zu Problemen kommen, wenn nur ein Screen offen ist und der ScreenModus verändert wird.
[Berichtet von Marcin Andrzejak]

Änd: Die UmgebungsVariable (SCREENCLOCK) kann nun maximal 1024 Zeichen lang sein. ←

38.23 (14.11.95):

Neu: Optionen: CPU=CpuUsage/T,CF=CpuFormat/K
Es kann nun die aktuelle CPU-Belastung in % angezeigt werden!

Release 1.73:

38.22 (17.07.95):

Neu: Option 'DoubleTicks/T' -> die 1/50 seks. verdoppeln, so daß es 1/100 sind ←

.
Nur in Verbindung mit 'ShowTicks/T'. Default ist OFF. [Wunsch von Raj Joshi] ←

Änd: Die Ticks werden nun 2 oder 3 stellig (wenn DoubleTicks = ON) angezeigt.

Release 1.72:

38.21 (20.06.95):

Neu: Option: 'ShowTicks/T' -> 1/50 seks anzeigen? [Wunsch von Raj Joshi]
Mit Einschränkung, siehe ST=ShowTicks/T
Default ist OFF.

Fix: Nachdem ein Notify-Signal oder Ctrl-C empfangen wurde (Env-Var oder Locale ←
wurde verändert), wird nicht mehr gewartet.

Fix: "Titelleisten-flickern" vermindert

Release 1.71:

38.20 (30.04.95):

Änd: In der Env-Var können nun mehrere Zeilen verwendet werden.

38.19 (26.04.95):

Änd: Es muß nun Scr.defaultTitle UND Scr.title NIL sein, damit nichts ←
angezeigt wird.
[Wunsch von Rodney Hester, da es sonst mit Klondike V1.8 nicht läuft!]
Wenn .defaultTitle = NIL und .title # NIL, dann wird .title zum Matchen ←
benutzt.

Release 1.7:

38.18 (04.03.95):

Dok: Node Was macht das Programm? neu geschrieben.

Dok: Node Wie kann man mich erreichen? verbessert.

Fix: Unschöne Anzeigen, bei der der alte Text nicht richtig gelöscht wurde.

[Berichtet von J.Frede]

[Ich hatte nur ein "<" anstatt ein ">" benutzt;-]

Änd: Unter OS v38, wird mit den Farben 0 auf 1 gemalt.

Das sollten nun endgültig die richtigen sein;-)

Falls es immer noch nicht klappen sollte, werde ich in der nächsten
Version wohl 2 neue Optionen (Front-, BackPen) einbauen;-)

[Berichtet von T.Schwarz]

Release 1.6:

38.17 (01.03.95):

Änd: Fehlermeldungen sind nun nur noch Nummern!

Dok: Mögliche Fehlermeldungen erklärt.

38.16 (26.02.95):

Fix: Bei OPS=ON, wurden keine PubScreens erkannt. [berichtet von Hanno]

Release 1.5:

38.15 (16.02.95):

Änd: Es wird nun nur noch direkt um Int.PrintIText() der BarLayer ge- bzw. ←
unlocks.

Dok: Start von der Workbench erklärt.

Dok: Piktogramm beigelegt!

38.14 (15.02.95):

Änd: Unter OS v38, wird als Hintergrund die Farbe 2 und als Vordergrund die 1 ←
benutzt.

Das sollte nun endlich funktionieren!?

Fix: Falls ein AutoScroll-Screen mit LeftEdge = 0 benutzt wird, wurde die x ←
pos.

so berechnet, als obs ein normaler Screen wäre.

38.13 (14.02.95):

Neu: Falls der Layer und der PubScreen am Ende des Programms gelockt waren,
werden sie entlockt.

Das kann (!) z.B. passieren, wenn ScreenClock während der Vorbereitungen
zum Anzeigen des Datums durch ein Fehler unterbrochen wurde.

Das sollte die seltenen 'Hänger' vermeiden!?

38.12 (13.02.95):

Neu: Ein paar NIL-Checks eingebaut.

Neu: Der aktuelle Screen wird vor dem Anzeigen, nochmals mit dem Screen
verglichen, der am Anfang der Berechnungen aktuell war.

Dadurch sollte das Programm bei nicht-PubScreens sicherer sein.

38.11 (12.02.95):

Änd: Unter OS v38, wird nun der ShadowPen anstatt der ShinePen als
Hintergrund benutzt.

38.10 (11.02.95):

Neu: Option 'AV=AlwaysVisible' -> immer im sichtbaren Bereich?

Default ist ON!

38.9 (09.02.95):

Neu: Die Anzeige erscheint nun immer im sichtbaren Bereich!
Wichtig für überbreite (AutoScroll) Screens!
Dok: Beispiele zur Option Screens

Release 1.4:

38.8 (29.01.95):

Fix: Es wurden auch Screens benutzt die zwar einen Titel aber keine Titelleiste hatten!
Änd: Der Text wird nun mit Intuition.PrintIText() geschrieben, so daß seltene Farbveränderungen der Leiste nun nicht mehr vorkommen sollten, da der Front- und BackPen von dieser Proz. gesetzt wird!
Änd: Dadurch gibts wieder nur noch eine Version (für OS 2.1)! Es wird auch nicht mehr auf die richtige OS Version geprüft!
Fix: FreeScreenDrawInfo() wurde nicht gemacht;-(
Änd: Es wird vorm Schreiben nicht mehr geprüft, ob ein nicht-öffentlicher Bildschirm noch da ist!

Release 1.3a:

38.7 (28.01.95):

Fix: Es wird geprüft, ob der BarLayer schon ge'lock't ist.
Wenn ja, dann wird nicht gewartet bis er frei ist!

Release 1.3:

38.6 (27.01.95):

Neu: 'OPS=OnlyPubScreens/T' -> es werden nur PubScreens benutzt
Neu: Falls der aktuelle Screen ein PubScreen ist, wird er während des Schreibens gelockt, damit er nicht geschlossen werden kann! Wenn nicht, wird vor Gfx.Text() nachgeguckt, ob er noch da ist!
Fix: Eine 'unschöne' Anzeigen sollten nun nicht mehr vorkommen!
Neu: 'Pri=TaskPriority/K/N' -> Priorität des Task. -3 <= pri >= +3
Neu: Notify auf 'ENV:sys/locale.prefs'

38.5 (26.01.95):

Fix: Es wird nun der Screen.barLayer vor dem Schreiben gelockt.

Release 1.2:

38.4 (13.01.95):

Neu: Es sollten nun die richtigen Pens (wie der ScreenTitel) benutzt werden!
Neu: Zwei verschiedene Versionen für OS2.x (>=v38) und OS3.x (>=v39) Es wird beim Start auf die richtige OS Version geprüft!
Fix: CloseLocale() wurde am Ende des Progs vergessen;-(
Fix: Locale.FormatDate() hängt bei manchen Formaten (die abgekürzten!) ein 0X zuviel, dadurch fehlte bei den anderen das letzte Zeichen des Datums!
Fix: Falls das Notify nicht klappte, wurde am Ende des Programms trotzdem Dos.EndNotify() aufgerufen.
Ein paar Aufräumarbeiten.

Release 1.1:

38.3 (12.01.95):

Fix: Das Neueinlesen der Parameter klappte nicht so gut.
Neu: Bei MF = "" oder DF = "" werden die defaultFormate wieder gesetzt.

38.2 (11.01.95):

Neu: Die EnvVar wird bei Änderungen automatisch eingelesen!
Änd: Option 'Memory' ist nun eine Toggle-Option, s. doku

38.1 (09.01.95):

Änd: Locale.FormatDate() anstatt Dos.DateToStr() und SPrintf().

Daher auch der Aufstieg zur Version 38!
Änd: 'Seconds/S' fällt deswegen weg.
Neu: Option 'DF=DateFormat/K' -> Format der Datums-/ZeitAnzeige
Neu: Option 'MF=MemoryFormat/K' -> Format der SpeicherAnzeige
Neu: Environmentvariable (ScreenClock) für die Parameter

Release 1.0:

37.3 (05.01.95):

Neu: Option 'Memory/S' -> zeigt auch den Speicher an
Änd: Einige interne Änderungen -> weniger Zeitverbrauch !?

37.2 (03.01.95):

Neu: Option 'Screen/K' - gibt ein Pattern für die Bildschirmtitel
an die benutzt werden sollen!
Neu: Es wird ein MsgPort eingerichtet, zum überprüfen, ob
ScreenClock schon läuft.

37.1 (02.01.95):

Angefangen, da ich bemerkte, daß KCX bei mir nur deswegen läuft;-)

Erklärung der Abkürzungen:

Neu: Es wurde etwas neues eingebaut.
Änd: Es wurde etwas geändert.
Fix: Es wurden Fehler verbessert.
Dok: Es wurde etwas in der Anleitung geändert/verbessert.

1.21 ScreenClock/Wie kann man mich erreichen?

Falls ihr Verbesserungsvorschläge, irgendwelche Fragen, Bugreports usw.
habt, dann könnt ihr mir schreiben, am besten per e-mail!

Z-Netz: T.Igracki@BAMP.berlinet.de
FIDO : Thomas_Igracki%2:2410/309.34
UseNet: lokai@cs.tu-berlin.de -oder-
t.igracki@bamp.berlinet.de (bevorzugt!)
Post : Thomas Igracki, Siegenger Str. 34, Germany-13583 Berlin

Falls ihr mir eine kleine (oder auch große!) Geldspende für meine Arbeit an
ScreenClock zukommen lassen wollt, dann schickt es mir in Bar oder überweist
es mit einem entsprechenden Vermerk (und EMail) auf folgendes Konto:

Empfänger: Igracki, Thomas
KontoNr. : 860169138 BLZ: 100 500 00
bei : Sparkasse Berlin

Vermerk : Spende für ScreenClock, <Dein Name + EMail>

Wenn ihr mir dann noch eine eMail schickt, werde ich euch informieren,
wenn eine Version von ScreenClock fertig ist! Und, wenn gewünscht
automatisch zumailen!

Schreibt mir aber wenigstens eine eMail, wenn ihr ScreenClock benutzt!

Viel Spaß mit dem Programm!

1.22 SysInfo.library by Petri Nordlund

SysInfo.library
~~~~~

SysInfo.library was developed to bring together all the different utility programs that add some new features to Amiga task handling, like CPU usage calculation. SysInfo.library was first developed for Executive, but it's possible to rewrite it to support other similar programs. It's not necessary to support all SysInfo.library functions in all implementations.

Please consult the file 'SysInfo.doc' for more information.

The provided examples can be compiled with GCC or SAS/C.

You must copy the libs/SysInfo.library to your libs: directory.

Copy the include files to appropriate directory:

GCC: GCC:os-include/  
SAS/C: INCLUDE:

If you're using GCC, you must also copy the files in gcc/lib/ to GCC:lib/ and files in gcc/os-include/ to GCC:os-include/.

---