3.

1.

Multics (Multiplexed Information and Computing Service).

Wann wurde das Betriebssystem entwickelt? Wird es noch weiterentwickelt?

Erst in 1964 von General Electric & Bell Labs entwickelt, in 1970 an Honeywell verkauft worden. Wird nicht mehr weiterentwicklet. Wurde im 2000 zum letzten Mal benutzt.

Hat das Betriebssystem einen besonderen Anwendungszweck?

Das Betriebssystem wurde mit komerziellen Zwecken entwickelt, also nichts besonders.

Ist eine spezielle Hardware nötig? Braucht es ja!

Da es für den normalen Benutzer auch nutzlich sein musste, brauchte es keine spezielle Hardware.

Stellt es bestimmte Dienste (eng: services) bereit?

Ja, es implementiert ein einziges Storage für Datazugriff, wo es nicht mehr einen klaren Unterschied zwischen Files und Storage memories gibt. Der "Dynamic Linker" wird auch eingeführt, was, mit dem Vortel für den Benutzer, dass gelinked Bibliotheken aktualisiert werden können, und potentielles Sparen an Speicherplatz.

Worin hebt es sich von anderen Betriebssystemen ab?

Es ist das erste System mit hierarchischem Data System, und dass ein Stack pro Prozess im Kernel benutzt hat, und ein getrenntes Stack für jeden Ring. Ausserdem wurde beim Schreiben von Multics viele Emphasis auf IT Sicherheit gemacht, was bisher nicht das wichstigste war.

Rechercheweg: Suche in Google, wobei 3. Ergebnis die ofizielle Seite von MIT -http://web.mit.edu/multics-history/ -für Multics ist. Hier steht aber nur eine Zusammenfassung über ihre Geschichte. Auf dieser Seite habe ich einen Link zu http://www.multicians.org/features.html, eine Seite wo Multics fans die ganze Information bezüglich des nicht mehr benutzten Betriebssystem gesammelt haben.

EROS

Wann wurde das Betriebssystem entwickelt? Wird es noch weiterentwickelt?

EROS (Extremely Reliable OS) wurde von 1991-2005 an University of Pennsylvania und dann an der Johns Hopkins University entwickelt. Entwicklung ist eingestellt, es gibt zwei aktive Nachfolgerprojekt CapROS und Coyotos.

Hat das Betriebssystem einen besonderen Anwendungszweck?

EROS und seine Nachfolger sind besonders für Anwendungen geeignet, bei denen Sicherheit oder Zuverlässigkeit eine besondere Rolle spielen. EROS basierte Systeme haben Anwendung gefunden für besonders sichere Window Systems und Network Stakes. Außerdem bieten sie zusammen mit der Programmiersprache BitC komplette formale Programmverifikation auf Systemebene.

Ist eine spezielle Hardware nötig?

Nein, EROS läuft auf normaler Commodity-Hardware.

Stellt es bestimmte Dienste (eng: services) bereit?

EROS bietet keine besonderen Dienste, sondern nur eine besonders sichere und zuverlässige Architektur.

Worin hebt es sich von anderen Betriebssystemen ab?

1/2

EROS unterstützt insbesondere automatische Daten und Prozess Persistenz und Capability-basierte Rechtevergabe. Anwendungen werden in kleine Komponenten zerlegt, welche nur über auf Kernel-Ebene definierte Schnittstellen miteinander kommunizierende können. Jede Komponente hat beschränkte Capabilities (zu Vergleichen mit Zugriffsrechten in UNIX), welche ihr erlauben oder verbieten über IPC (Interprozesskommunikation) auf eindeutig definierte Schnittstellen anderer Komponenten zuzugreifen. Sicherheit wird durch sogenanntes Confinement gewährleistet, bei dem die Propagation der Capabilities beschränkt wird.

(Wikipedia)[http://en.wikipedia.org/wiki/EROS_(microkernel)]

(EROS homepage)[http://www.eros-os.org]

Plan9

Wann wurde das Betriebssystem entwickelt? Wird es noch weiterentwickelt?

Bell Labs begann 1987 mit der Entwicklung von Plan9. Es wurde hauptsächlich zu Forschungszwecken als Unix Nachfolger entwickelt. Seit der Veröffentlichung der vierten Edition 2002 ist Plan9 Open Source. Plan9 wird von einer kleinen, aber nochh aktiven Gruppe, weiterhin genutzt und gepflegt. Die aktivere Entwicklung findet allerdings wohl bei Forks, wie z.B. Plan9Front statt.

Hat das Betriebssystem einen besonderen Anwendungszweck?

Plan9s Entwicklung diente vor allem dazu, Weiterentwicklungen vom originalen Unix Modell zu erforschen. Es sollte neue Technologien, sowie Designkonzepte aufzeigen und ist nicht für den typischen Endanwender geeignet. Treiber und Anwendungen sind nur wenig vorhanden. Plan9 dient heute hauptsächlich als Forschungsplattform für Betriebssystementwicklung.

Ist eine spezielle Hardware nötig?

Downloads für installierbare Systeme sind derzeit nur für x86-basierende Hardware und den Raspberry Pi erhältlich. Es existiert eine Hardware-Kompatibilitätsliste auf der Plan9 Wiki. Außerdem besteht die Möglichkeit Plan9 auf einer Reihe von virtuellen Systemen laufen zu lassen. Es gibt verschiedene historische Ports auf andere Hardware, wie z.B. auf PowerPC oder SPARC.

Stellt es bestimmte Dienste (eng: services) bereit?

Plan9 bietet keine besonderen Dienste, allerdings eignet sich Plan9 für "distributed computing" (Verteiltes Rechnen) Mit Plan9 können Computing Grids erzeugt werden, wobei ein virtueller Supercomputer aus einem lose gekoppeltem Computer Cluster erzeugt wird.

Worin hebt es sich von anderen Betriebssystemen ab?

Plan9 treibt das "Everything is a file" Konzept von Unix weiter voran. So kann auf viele Ressourcen (z.B. Dokumente, aber auch Hardware Geräte) über das Dateisystem zugegriffen werden. Plan 9 übernahm dieses Konzept für "distributed computing" mit 9P (Plan 9 Filesystem Protocol).

(Wikipedia) [http://en.wikipedia.org/wiki/Plan_9_from_Bell_Labs]

(Plan 9 Wiki) [http://www.plan9.bell-labs.com/wiki/plan9/plan_9_wiki]

2.a)

```
#include <stdio.h>
```

```
int main ()
{
	float f;
	int i;
	long l;

	i = 1024;
	l = (long) i;
	printf("%d = %ld \n",i,l); // 1024 = 1024
	// longs und ints sehen gleich aus, Typkonvertierung macht Sinn.

	l = 23;
	printf("l = %ld\n",l); // l = 23
	// Werte können überschrieben werden.

	i = (int) l;
	printf("%ld = %d\n",l,i); // 23 = 23
	// Umgekehrt können longs auch in ints umgewandelt werden.

	l = 17179869184;
	printf("l = %ld\n",l); // l = 17179869184
	// longs können sehr groß sein.

	i = (int) l;
	printf("%ld = %d\n",l,i); // 17179869184 = 0
	// Wenn ein zu langer long in einen int gecastet wird, kriegt
	// dieser den Wert 0.

	f = 5/2;
	printf("5/2 = %6.2f\n",f); // 5/2 =   2.00
	// Wenn Zahlen nicht explizit als floats eingegeben werden,
	// dann wird int-Division verwendet.

	f = (float)5/2;
	printf("5/2 = %6.2f\n",f); // 5/2 =   2.50
	// Selbst wenn Zahlen nach der Division in einen float gecastet
	// werden, dann ist das Ergebnis eine float-Division.

	f = 1.333;
	i = (int)f;
	printf("1.333 = %d\n",i); // 1.333 = 1
	// Wenn floats in ints gecastet werden, geht der Nachkommaanteil
	// verloren.

	f = 5.0/2;
	i = (int)f;
	printf("2.5 = %d\n",i); // 2.5 = 2
	// Beim int-casting wird nicht gerundet.

	return 0;
}
```

ja und weiter?

2.bc)

```
// Tobias Lohse, Sven Klaus, Luisa Castano Jurado

#include <stdio.h>

double cel2fahr(double cel)
{
	return 1.8 * cel + 32.0;
}

int main()
{
	printf("   C     |   F \n");
	printf("---------|---------\n");
	for (double cel=-30; cel<=100; cel+=10)
	{
```



```
		printf("%6.1f   |  %5.1f \n", cel, cel2fahr(cel) );
	}

	return 0;
}
```

Code abgabe? zu spät...

2/3