1. Beschreibe kurz den grundsätzlichen Unterschied zwischen NAS und SAN

NAS und SAN sind beides Lösungen für Netzwerkspeicherung. Bei NAS ist der Speicher direkt mit einem File Server verbunden, das den Speicher auf file-level für andere Benutzer verfügbar macht.   
SAN macht den Speicher auf einem tieferen block-level für andere Benutzer verfügbar, was die file system aufgaben den Benutzern überlässt

1. Üblicherweise werden Disks formatiert. Dies hat bei Festplatten zur Folge, dass in jedem Sektor Platz für einen Trailer und Header benötigt werden. Wofür wird dieser Overhead an genutztem Speicher gebraucht? Inwiefern hat dies Einfluss auf die Grösse des Nutzdatenblocks bzw. was sind Gründe für die Wahl einer unterschiedlichen Grösse des Nutzdatenblocks?

Trailer und Header bezeichnen den Anfang, beziehungsweise das Ende eines Sektors, wobei der Trailer eine Checksumme enthält, die die Integrität der vorangegangenen Daten überprüft.

Relativ unklare zweite Frage, aber unter der Annahme, dass die Wahl zwischen zB 256, 512 und 1024 Bytes bei der low-level formatierung gemeint ist wäre die Antwort, dass so zwar weniger/mehr Sektoren auf einen Track passen, dafür aber auch weniger/mehr Trailer und Header (und auch beabsichtigte Lücken) benötigt werden, was den Nutzdatenblock erhöht/verringert.

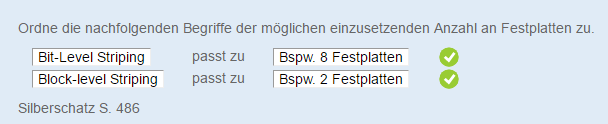
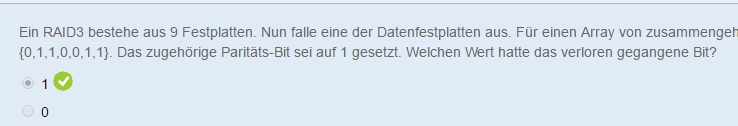
1. Was ist Sector Sparing? Wozu wird es benötigt? (0.5) Wie unterscheidet sich Sector Sparing von Sector Slipping? (0.5) Was ist der Nachteil von Sector Sparing gegenüber Sector Slipping? (0.5)

Sector Sparing, auch Hot FIxing genannt verhindert, dass Daten in einen defekten Sektor gespeichert werden. Das geschieht dadurch, dass das Betriebssystem einen defekten Sektor erkennt und die zu speichernden Daten in einen sicheren, freien, dafür vorgesehen Ort speichert. Dabei werden dem Nutzer/den Anwendungsprogrammen keine Fehlermeldungen (wie fail oder abort) gesendet.

Angenommen wir haben die Sektoren 1-7 und freien Platz: 1 2 3 4 5 6 7 S S (wobei S einen freien/spare Sektor bezeichnet). Wenn dann Sektor 3 beschädigt würde, würde nun Sektor 3 auf einen dieser freien Sektoren gemappt: 1 2 3 C 4 5 6 7 3 S (wobei C einen defekten Sektor bezeichnet)

Bei Sektor Slipping würde nun über diesen defekten Sektor hinweg gemappt, also quasi alles darüber hinweg verschoben. Nach dem Sektor Slipping würde unser Beispiel so aussehen:   
1 2 C 3 4 5 6 7 S

Der Nachteil von Sparing liegt darin, dass zuerst zu diesem gemapptem Sektor gefunden werden muss, um an die darin gespeicherten Daten zu gelangen, während bei Slipping der gesuchte Sektor einfach ein Sektor später kommt.

1. 
2. 
3. Wie funktioniert der Zugriff eines Hauptspeicherprozesses auf ein Memory Mapped File (0.5)? Wie ermöglicht dies das Teilen von Informationen zwischen Prozessen (0.5)?

Die Pages können in den virtuellen Speichere "gemappt" werden. Das heisst, sie werden nicht effektiv dorthin gelagert, sondern mehr von dort verlinkt. Das können nun natürlich mehrere Prozesse machen, wodurch die Pages so für mehrere Prozesse verfügbar sind, da sie ja nicht wirklich im Hauptspeicher für einen einzigen Prozess liegen, sondern eben nur dort gemappt sind.

1. RAID-3 ist ein Vorgänger der RAID-5 und ist nicht in der Praxis verbreitet, obwohl es zur Reduzierung von Kosten dienen sollte. Begründen Sie warum RAID-5 eine bessere Option ist. Wie sind die zwei RAID Implementationen ähnlich?

RAID 3 und 5 brauchen grundsätzlich beide ein Paritätsbit. Der Unterschied ist, dass bei RAID 5 dieses Paritätsbit auf alle Speicher verteilt ist, während bei RAID 3 die Paritätsbits alle auf einem Speicher sind. Dadurch wird dieser Speicher zum Flaschenhals. Durch das Verteilen dieser Paritätsbit wird dieser Flaschenhals-Effekt vermieden.

1. Was passiert wenn ein Benutzer A einen weichen Link auf eine dem Benutzer B gehörende Datei erzeugt und Benutzer B diese löscht? Was geschieht im Falle eines harten Links?

Üblicherweise führt eine Löschung der Datei dazu, dass der weiche Link von Benutzer A ins Nichts zeigt, da der weiche Link lediglich eine Referenz auf die Zieldatei ist und nicht gleichwertig zum eigentlichen Dateisystem-Eintrag der referenzierten Datei.

Im Falle eines Hard Links hätte A grundsätzlich weiterhin Zugriff auf diese Datei, da die eigentliche Datei erst dann wirklich gelöscht wird wenn der letzte Hard Link auf diese Datei gelöscht wird.

(Vergleiche für eine detailliertere Antwort die bereits eingereichte Lösung für die praktische Serie 4).